TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

Santeri Moberg
Joonas Hilke
Juuso Säärelä

Palvelunestohyökkäykseen osallistuvan IoT-laitteen havaitseminen tukiasemassa

Kandidaatintyö
Tietotekniikan tutkinto-ohjelma
Tammikuu 2019

Moberg S, Säärelä J, Hilke J. (2019) Palvelunestohyökkäykseen osallistuvan IoT-laitteen havaitseminen tukiasemassa. Oulun yliopisto, tietotekniikan tutkinto-ohjelma. Kandidaatintyö, 41 s.

TIIVISTELMÄ

Esineiden internet (Internet of things, IoT) tulee kasvamaan huomattavasti tulevaisuudessa ja markkinoille ilmestyy jatkuvasti heikolla tietoturvalla varustettuja IoT-laitteita. Palvelunestohyökkäyksiä tekevät bottiverkot ovat alkaneet suosimaan niitä ja ne koostuvatkin suurimmaksi osaksi saastuneista IoT-laitteista. Tällaisten IoT-laitteiden tietoturvan parantamiseen tarvitaan jatkuvasti uusia tietoturvaratkaisuja, joiden avulla voidaan ehkäistä palvelunestohyökkäyksiä ja siten suojata sekä internetin käyttäjiä että palveluita bottiverkkojen luomalta kasvavalta uhalta.

Työssä toteutettiin langattomassa tukiasemassa ajettava ohjelma, jonka tarkoitus on havaita palvelunestohyökkäykseen osallistuva tukiasemaan yhdistetty IoT-laite. Ohjelma suunniteltiin havaitsemaan UDP-, TCP SYN-, DNS- ja ICMP-tulva-hyökkäykset. Havaitseminen tapahtuu tarkkailemalla ja analysoimalla tukiasemaan yhdistyneiden IoT-laitteiden verkkoliikennettä. Havaittuaan hyökkäävän laitteen, ohjelma ilmoittaa hyökkäyksestä tallentamalla hyökkäykseen liittyvät tiedot paikalliselle verkkosivulle.

Ohjelmaa testattiin simuloimalla edellä mainittuja palvelunestohyökkäystyyppejä itsetehdyillä DoS-työkaluilla. Testien perusteella todettiin, että ohjelma pystyy onnistuneesti havaitsemaan kaikki testeissä simuloidut palvelunestohyökkäystyypit, vaikka ohjelman rajallinen suorituskyky vaikutti negatiivisesti ohjelman kapasiteettiin analysoida verkkoliikennettä. Lisäksi huomattiin, että ohjelma voi tulkita suuren määrän normaalia UDP-verkkoliikennettä palvelunestohyökkäykseksi.

Avainsanat: esineiden internet, bottiverkko, DDoS, hajautettu palvelunestohyökkäys.

Moberg S, Säärelä J, Hilke J. (2019) Detecting an IoT-device participating in a denial-of-service attack in an access point. University of Oulu, Degree Programme in Computer Science and Engineering. Bachelor's Thesis, 41 p.

ABSTRACT

Internet of things is expecting rapid growth and IoT-market is filled with devices that have poor cybersecurity. This in turn has resulted in botnets behind denial-of-service attacks to start favouring IoT-devices, thus today's botnets mainly consist of infected IoT-devices. These kinds of IoT-devices are in need of protection and different kinds of cybersecurity solutions are needed to ensure safety of our internet enabled society.

In this thesis we propose a program that detects if an IoT device is participating in a denial-of-service attack. The program runs in a wireless access point and it was designed to detect UDP, TCP-SYN, DNS and ICMP floods. The program detects these attacks by sniffing and analysing the internet traffic of the devices connected to the access point. When an attacking device is detected, the program displays information about the attack on a local web server.

Denial-of-service attacks of the aforementioned types were simulated with self-made attack tools. Based on tests, the program can detect all four of the attack types, though its limited performance affected negatively on its capability to analyse network traffic. The program can also mistake a large volume of legitimate UDP traffic as a denial-of-service attack.

Key words: internet of things, botnet, DDoS, DoS.

SISÄLLYSLUETTELO

TIIVISTELMÄ
ABSTRACT
SISÄLLYSLUETTELO
ALKULAUSE
LYHENTEIDEN JA MERKKIEN SELITYKSET
1. JOHDANTO 8
2. IoT-LAITTEET DDoS-HYÖKKÄYSTEN TAKANA 10
2.1. Palvelunestohyökkäys 10
2.1.1. Palvelunestohyökkäystyypit 10
2.1.2. Tunnettuja palvelunestohyökkäyksiä 11
2.1.3. Palvelunestohyökkäysten havaitseminen 12
2.1.4. DDoS palveluna 12
2.2. Bottiverkko 13
2.2.1. IoT-laitteen saastuttaminen osaksi bottiverkkoa 14
2.2.2. Tunnettuja bottiverkkohaittaohjelmia 14
3. TIETOLIIKENNEPAKETIT JA NIIDEN TARKKAILEMINEN 16
3.1. Tietoliikennepaketti 16
3.1.1. TCP (Transmission control protocol) 16
3.1.3. UDP (User datagram protocol) 17
3.1.4. ICMP (Internet control message protocol) 17
3.2. Pakettien tarkkaileminen 17
4. TYÖN KUVAUS 18
4.1. Käytetty laitteisto ja ohjelmisto 18
4.2. Havaitsinlaitteen turvallisuus 18
4.3. Ohjelman suunnittelu 19
4.4. Langattoman tukiaseman luominen 20
4.5. Ohjelman kuvaus 20
4.5.1. Verkkoliikenteen kuuntelu 21
4.5.2. Pakettien erittely ja analysointi 22
4.5.3. Verkkosivu 23
4.5.4. Ohjelman multiprosessointi 24
5. OHJELMAN TESTAUS 25
5.1. Testauksessa käytetty verkkoliikenne 25
5.2 Testausympäristö 26
5.3. Testaussuunnitelma 26
5.4. Testien tulokset 28

5.4.1. Testi 1 (DNS-tulva) 28
5.4.2. Testi 2 (TCP SYN-tulva) 29
5.4.3. Testi 3 (UDP-tulva) 29
5.4.4. Testi 4 (ICMP-tulva) 30
5.4.5. Testi 5 (Normaali verkkoliikenne: VLC media player) 31
5.4.6. Testi 6 (Normaali verkkoliikenne: Plex Media Server) 31
6. POHDINTA 32
6.1. DDoS-hyökkäyksen tunnistusheuristiikka ja kynnysarvot 32
6.2. DDoS-työkalut 33
6.3. Scapy-kirjaston heikko suorituskyky 33
6.4. Jatkokehitys 33
7. AJANKÄYTTÖ 35
8. YHTEENVETO 36
9. LÄHTEET 37

ALKULAUSE

Haluamme kiittää työn ohjaajaa Teemu Tokolaa sekä professori Juha Röningiä neuvoista ja palautteesta työn aikana. Haluamme kiittää myös Marko Laaksoa avusta työn aiheen rajaamisessa.

Oulu, Tammikuu 12. 2019

Santeri Moberg
Juuso Säärelä
Joonas Hilke

LYHENTEIDEN JA MERKKIEN SELITYKSET

ACK	TCP-paketti josta löytyy Acknowledgement lippu
ARP	Address Resolution Protocol
C&C	Command & Control
DDoS	Distributed-Denial-of-Service
DHCP	Dynamic Host Configuration Protocol
DNS	Dynamic Name Server
DoS	Denial-of-Service
FIN	TCP-paketti josta löytyy
FTP	File Transfer protocol
GIL	Global Interpreter Lock
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
IRC	Internet Relay Chat
JSON	JavaScript Object Notation
LTE	Long Term Evolution
MAC	Media Access Control
MITM	Man in the middle
NAT	Network Address Translation
NTP	Network Time Protocol
RAM	Random Access Memory
RFID	Radio Frequency Identification
SMTP	Simple Mail Transfer protocol
SYN	TCP-paketti josta löytyy Synchronization lippu
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

1. JOHDANTO

IoT eli esineiden internet tarkoittaa internet-verkon laaja-alaista leviämistä useisiin ympärillämme oleviin arkipäiväisiin esineisiin, joiden toimintaa voidaan ohjata tai seurata internetin yli [1]. Tulevaisuudessa esineiden internetiin kuuluvien laitteiden määrä tulee kasvamaan huomattavasti, mikä voidaan huomata useiden eri tutkimusyhtiöiden tekemistä tutkimuksista. Esimerkiksi markkinatutkimusyhtiö IDC:n vuonna 2014 tekemän tutkimuksen mukaan IoT-laitteiden määrä oli vuonna 2014 11,4 miljardia ja ennustetun kasvun mukaan se tulisi olemaan 25,2 miljardia vuoteen 2020 mennessä [2]. Kun taas tutkimus- ja konsultointiyritys Gartnerin vuonna 2017 tekemässä tuoreemmassa tutkimuksessa ennustettiin, että esineiden internet sisältää yli 20,4 miljardia laitetta vuoteen 2020 mennessä [3]. Tuoreemmasta ennusteesta voidaan nähdä, että vaikka kasvu ei olekaan niin suuri kuin aiemmin odotettiin, on IoT-laitteiden määrä tulevaisuudessa silti merkittävä.

IoT-laite voi olla esimerkiksi älyjääkaappi, ajoneuvo, WSN-laite, RFID-tunniste tai melkein mikä tahansa laite, joka on yhdistyneenä internetiin ja kommunikoi sen välityksellä. IoT-laitteet operoivat yleensä ilman ihmisen valvontaa ja kommunikoivat langattomien yhteyksien yli käyttäen erilaisia teknologioita kuten WLAN, 3G tai LTE [4]. Lähitulevaisuudessa Oulun yliopistossakin tutkittavat ja kehitettävät 5G ja 6G [5] parantavat huomattavasti reaaliaikaista tiedonsiirtoa ja suurta kaistanleveyttä vaativia käyttötapauksia. Kyseisten langattomien teknologioiden halventuminen ja tehonkulutus voi johtaa myös niiden yleistymiseen IoT-laitteissa. IoT-laite voi olla suoraan yhteydessä internetiin tai epäsuorasti toisen laitteen välityksellä. Epäsuoraa yhteyttä käytetään, jos IoT-laitteella on rajallinen tehonkäyttö tai laskentateho. Esimerkiksi RFID-järjestelmässä RFID-lukija on yhteydessä internetiin ja toimii yhdyskäytävänä RFID-tunnisteiden ja internetin välillä [6]. Jotta IoT-ympäristön laitteet voivat kommunikoida energiatehokkaasti, ne käyttävät yleensä WLAN-verkon lisäksi WPAN-verkon teknologioita kuten Bluetoothia ja Zigbeetä.

IoT-laitteiden valmistuksessa ei aina priorisoida tietoturvaa, vaan käytettävyys ja tuotteen myynnin nopea aloittaminen ovat usein tärkeämpiä seikkoja. Krebsonsecurity uutisoi vuoden 2017 maaliskuussa Dahuan valmistamista valvontakameroista ja videotallentimista, joista löytyi haavoittuvuus, joka antoi kenen tahansa ohittaa kirjautumisprosessin saadakseen laitteen hallintaansa [7]. Ongelma IoT-laitteiden tietoturvassa ei ole uusi, sillä esimerkiksi jo vuonna 2014 BBC uutisoi älyjääkaapista, joka osana yli sadan tuhannen laitteen bottiverkkoa lähetti roskaposteja 2013 vuoden loppupuolella [8]. Perinteiset laitteessa ajettavat tietoturvaratkaisut, kuten palomuurit ja erilaiset antivirusohjelmat, eivät sovi IoT-ympäristöön laitteiden rajallisen virrankulutuksen ja laskentatehon vuoksi [9]. Myös erilaisten IoT-laitteiden laaja kirjo tekee sopivien tietoturvaratkaisujen löytämisestä vaikeaa. Nämä seikat altistavat IoT-laitteet useiden erilaisten hyökkäysten kohteeksi [10].

Hajautetut palvelunestohyökkäykset (DDoS-hyökkäykset) ovat kasvava uhka, johtuen niiden helposta toteutuksesta ja työkalujen saatavuudesta. Palvelunestohyökkäysten torjumiseen erikoistuneen Akamain raportin mukaan DDoS-hyökkäysten määrä kasvoi 14%, kun verrattiin vuosien 2016 ja 2017 viimeisiä neljänneksiä [11]. Palvelunestohyökkäysten torjuminen on moninkertaisesti vaikeampaa, kuin niiden toteuttaminen. Näissä hyökkäyksissä käytettävät bottiverkot ovat alkaneet suosimaan IoT-laitteita, johtuen niiden suuresta määrästä ja heikosta tietoturvasta. Suurin osa IoT-laitteista on myös aina päällä, jolloin ne ovat jatkuvasti hyökkäysvalmiudessa.

Usea eri valmistaja on huomannut IoT-laitteiden luoman tietoturvauhan ja alkanut tarjoamaan erilaisia tietoturvalaitteita, jotka suojaavat IoT-systeemejä hyökkäyksiltä [12,13,14]. Nämä laitteet ovat yleensä erilaisia langattomia tukiasemia, joihin IoT-laitteet ovat yhdistyneenä. Tietoturvalaitteet valvovat IoT-laitteiden toimintaa ja käyttävät erilaisia tietoturvamenetelmiä niiden turvaamiseksi.

Tässä työssä esitellään ohjelma, jonka tarkoitus on havaita saastunut IoT-laite sen osallistuessa palvelunestohyökkäykseen. Ohjelmaa ajetaan Raspberry Pi 2 -tietokoneella, joka toimii langattoman verkon tukiasemana IoT-laitteille. Sitä voidaan myös ajaa reitittimillä, joissa on Linux-pohjainen käyttöjärjestelmä. Ohjelma tarkkailee läpikulkevaa tietoliikennettä, ja ilmoittaa paikallisella verkkosivulla, mikäli tukiasemaan yhdistynyttä IoT-laitetta epäillään palvelunestohyökkäykseen osallistumisesta.

2. IoT-LAITTEET DDoS-HYÖKKÄYSTEN TAKANA

Hajautetussa palvelunestohyökkäyksessä uhrin IP-osoitteeseen ohjataan suuri määrä verkkoliikennettä useasta eri kohteesta, jolloin normaali tiedonsiirto uhrin ja muun internetin välillä estyy. DDoS-hyökkäyksellä on monia eri käyttökohteita ja siitä on muodostunut internetissä myytävä palvelu. Tyypillisesti palvelunestohyökkäykset toteutetaan hajautetusti käyttämällä bottiverkkoja, mutta on olemassa myös muita hyökkäystapoja, kuten viime aikoina otsikoissa ollut vahvistushyökkäys käyttäen memcached-palvelimia.

Bottiverkko on joukko internetiin yhdistyneitä laitteita, joiden tietoturva on murrettu ja hallinta siirtynyt kolmannelle osapuolelle. Bottiverkot koostuvat suurimmilta osin IoT-laitteista. IoT-laitteen saastuttaminen osaksi bottiverkkoa tapahtuu tyypillisesti haittaohjelmilla. Tunnetuimpia bottiverkkohaittaohjelmia ovat Bashlite, Mirai, Brickerbot ja Reaper.

2.1. Palvelunestohyökkäys

Palvelunestohyökkäysten tarkoituksena on häiritä hyökkäyksen kohteen verkkoliikennettä ohjaamalla sitä kohti valtava määrä liikennettä, joka estää normaalin tiedonsiirron kohteen ja muun internetin välillä [15]. Hyökkäys tapahtuu useimmiten useasta eri saastuneesta laitteesta yhtä aikaa, jolloin hyökkääjän lähettämien pakettien määrä moninkertaistuu suhteessa saastuneiden laitteiden määrään. Palvelunestohyökkäysten toteuttamisessa hyödynnetään useita eri tiedonsiirtoprotokollia ja hyökkäysvektoreita. Etenkin päivittämättömät laitteet tai uudet, ei yleisessä tiedossa olevat, haavoittuvaisuudet mahdollistavat suuren osan palvelunestohyökkäyksistä.

Hyökkääjä voi myös moninkertaistaa hyökkäyksessä käytetyn kaistanleveyden hyväksikäyttämällä haavoittuvia verkkopalvelimia, jolloin hyökkäystä kutsutaan vahvistinhyökkäykseksi. Vahvistinhyökkäyksissä hyökkääjä lähettää suhteellisen pienen määrän kyselyjä, jollekin haavoittuvalle verkkopalvelimelle, joka vastaa sille tulleisiin kyselyihin bittimääriltään huomattavasti suuremmilla vastauksilla. Näin hyökkääjä saa niin sanotusti peilattua verkkoliikenteen uhrille väärentämällä lähetettyjen kyselyiden IP-osoitteet uhrin IP-osoitteella [16].

Eri protokollat omaavat erikokoisia vahvistuskertoimia, jotka perustuvat palveluille suunnattujen tiettyjen kyselyjen vastausten kokoon. Vahvistinhyökkäyksissä käytetään tyypillisesti hyväksi UDP-pohjaisia palveluita, jotka eivät vaadi minkäänlaista kättelyä, vaan lähettävät vastauksen suoraan kyselystä löytyvään IP-osoitteeseen. [17]

2.1.1. Palvelunestohyökkäystyypit

Viestintäviraston mukaan palvelunestohyökkäykset voidaan jaotella kolmeen eri kategoriaan: Volumetrisiin hyökkäyksiin, protokollahyökkäyksiin ja sovellustason hyökkäyksiin [18]. Volumetristen hyökkäysten kategoriaan kuuluvat esimerkiksi UDP- ja ICMP-tulvat sekä muut lähetysosoitevääränettyt tulvat, joiden tarkoituksena on tukkia hyökkäyksen kohteelle varattua kaistanleveyttä. Varsinkin UDP-pohjaiset hyökkäykset ovat olleet suosittuja protokollan yhteydettömän luonteen takia.

Protokollahyökkäykset, kuten SYN, Ping of Death tai Smurf, pyrkivät kuluttamaan palvelinten tai verkkolaitteiden resursseja; hyökkäyksen kohteille pyritään

aiheuttamaan suuri määrä prosessointikuormaa, joka johtaa palvelimen tai verkkolaitteen kaatumiseen [19]. Sovellustason hyökkäykset hyväksikäyttävät esimerkiksi HTTP-, SMTP- tai FTP-protokollia tai “low-and-slow”-hyökkäyksiä. Volumetrisiin tai protokollahyökkäyksiin verrattuna “low-and-slow”-hyökkäykset eivät vaadi paljoa kaistaa, mutta kuluttavat palvelimen tai siinä toimivan palvelun resursseja näennäisesti aitojen kyselyjen, kuten HTTP GET/POST avulla [20].

Kaspersky Labin vuoden 2017 viimeisen vuosineljänneksen palvelunestohyökkäyksiin keskittyvän raportin mukaan kaikista hyökkäyksistä 55,63% oli TCP SYN-hyökkäyksiä, 15,24% oli UDP-pohjaisia hyökkäyksiä, ja 3,37% ICMP-hyökkäyksiä [21]. Vuoden 2018 ensimmäisen vuosineljänneksen raportin mukaan kaikista havaituista hajautetuista palvelunestohyökkäyksistä volumetriseen hyökkäystyyppiin kuuluvien UDP-hyökkäysten osuus oli 13,2%, protokollahyökkäyksiin kuuluvien TCP-SYN-hyökkäysten osuus oli 57,3% ja ICMP-hyökkäysten osuus 6,1% [22]. Toisen vuosineljänneksen raportista [23] voidaan nähdä, että varsinkin TCP SYN -hyökkäysten määrä kasvoi suuresti noin 23 prosenttiyksikköä ja UDP-hyökkäysten määrä pieneni 2,6 prosenttiyksikköä edelliseen vuosineljännekseen verrattuna. ICMP-hyökkäysten osuus puolestaan pieneni 1,5 prosenttiin. Nämä kolme hyökkäystyyppiä kattoivat yhteensä 92,3% kaikista hyökkäyksistä toisella vuosineljänneksellä.

2.1.2. Tunnettuja palvelunestohyökkäyksiä

Marraskuussa 2016 DDoS-hyökkäys kaatoi Lappeenrannassa kahden kiinteistön lämmitystä ohjanneen tietokoneen. Hyökkäyksen kohteeksi joutuneen tietokoneen automaattinen järjestelmä yritti korjata vikatilanteen käynnistämällä itsensä toistuvasti uudelleen, jonka seurauksena lämmitysjärjestelmä oli kiinni yli viikon. [24, 25].

Samassa kuussa WikiLeaks joutui DDoS-hyökkäyksen kohteeksi julkaistuaan Hillary Clintonin presidentinvaalikampanjan puheenjohtajan John Podestan Gmail-tililtä vuodettuja sähköpostiviestejä [26]. Hyökkäyksen tarkoitus oli todennäköisesti rajoittaa sähköpostien leviämistä internetissä.

DDoS-hyökkäyksiä käytetään myös kiristämiseen uhkaamalla uhria DDoS-hyökkäyksellä lunnaita vastaan. Useat ryhmät, jotka toimivat nimellä DD4BC (DDoS for Bitcoins), kiristivät useita internetissä toimivia viihde- ja finanssialan yrityksiä DDoS-hyökkäyksillä vuosina 2014-2016. Ennen hyökkäyksen alkamista, kohteelle saapui sähköposti, jossa yritystä pyydettiin siirtämään tietylle tilille vaihteleva summa Bitcoineja välttyäkseen palvelunestohyökkäykseltä. Hyökkäykset olivat useimmiten sovellustason hyökkäyksiä, mutta ryhmät hyödynsivät myös volumetrisia peilaushyökkäyksiä kiristysyrityksissään [27, 28].

Githubiin kohdistettiin 28. helmikuuta 2018 suurin siihen mennessä mitattu DDoS-hyökkäys [29]. Hyökkäyksen huippu oli parhaimmillaan 1,35 Tbps. Hyökkäyksen huomattuaan Github reititti liikenteen Akamai Prolexicin puhdistuspalvelimille, missä tuleva liikenne suodatettiin pahantahtoisista paketeista. Hyökkäyksessä käytettiin uudenlaista vahvistinhyökkäystyyppiä, joka käyttää hyväkseen internetissä olevia memcached-palvelimia. Kesti vain neljä päivää, kunnes Githubiin tehdyn hyökkäyksen ennätysmäinen koko rikottiin samaa menetelmää käyttävällä 1,7 Tbps kokoisella hyökkäyksellä [30]. Molempien ennätyksiä rikkoneiden hyökkäysten DoS-liikenteen pakettien tietosisällöstä löydettiin 50 Monero-kryptovaluutan lunnasvaatimus, joka vastasi hyökkäysten tekohetkellä noin 13000 euroa [31]. Cloudflaren tekemässä testissä

memcached-palvelin palautti 15 tavun kokoiseen kyselyyn 134 kilotavun vastauksen, mikä on lähes kymmentuhatkertainen vahvennus hyökkäyksen kaistanleveydelle [32]. Lisäksi memcached-palvelimien lähettämä liikenne on UDP-verkkoliikennettä, mikä soveltuu hyvin käytettäväksi palvelunestohyökkäyksissä johtuen sen yhteydettömästä luonteesta.

2.1.3. Palvelunestohyökkäysten havaitseminen

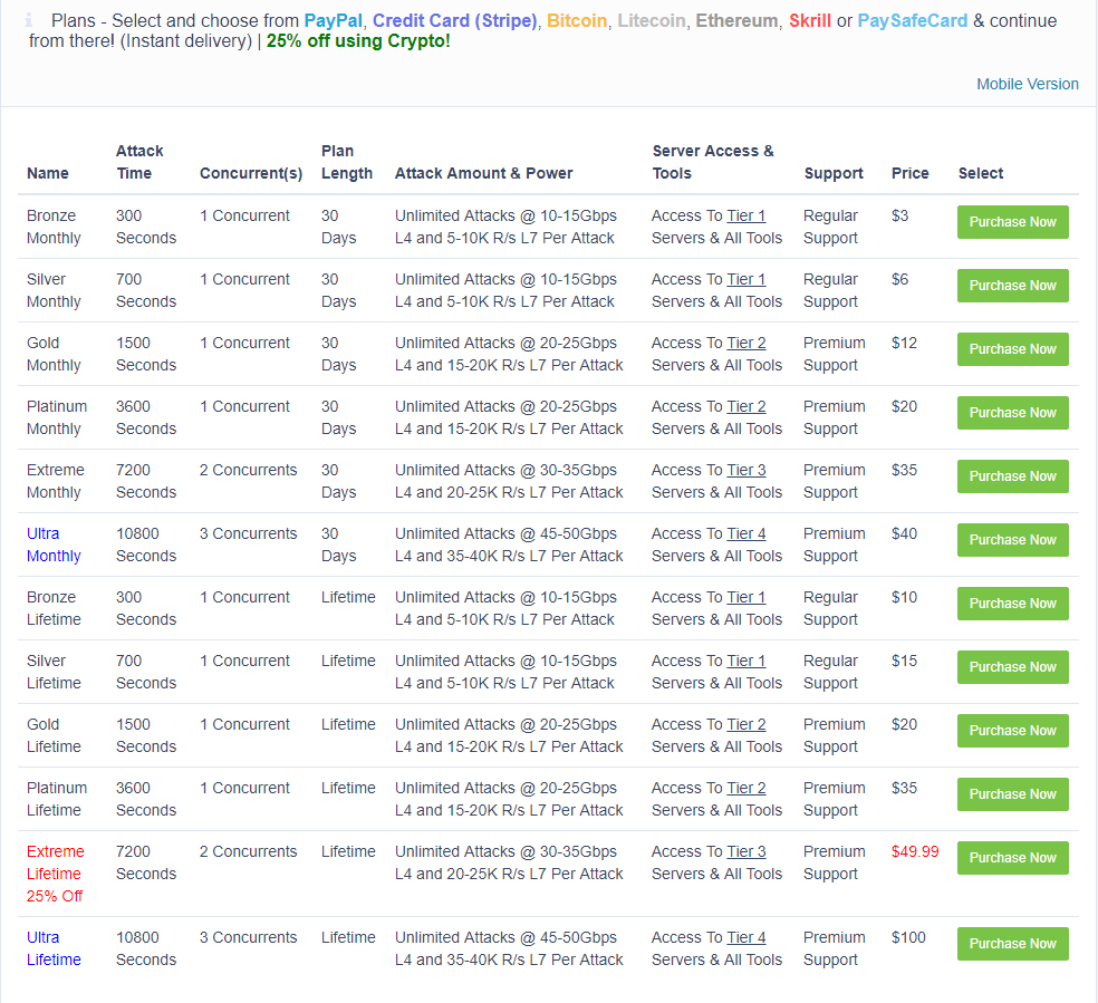
Palvelunestohyökkäyksien havaitsemiseen on kehitetty erilaisia ratkaisuja, joista monet keskittyvät hyökkäyksen havaitsemiseen uhrin järjestelmässä. Monet kyseisistä ratkaisuista käyttävät entropia- tai kaaosteoria-pohjaista heuristiikkaa hyökkäysten havaitsemiseksi. Nychis ym. kehittivät hyökkäysten tunnistamiseen entropia-pohjaisen heuristiikan, joka mittaa entropian määrää paketeista löytyvien lähettäjän ja vastaanottajan IP-osoitteista, porteista, yksittäisten tietoliikenneyhteyksien pakettimäärässä, ja verkosta löytyvien IP-osoitteiden kanssa tiedonsiirrossa olleiden vastaanottaja- ja lähettäjä-IP-osoitteiden määrässä [33].

2.1.4. DDoS palveluna

DDoS palveluna tarkoittaa palvelunestohyökkäyksien myymistä internetissä. Venäläinen tietoturvayhtiö Kaspersky Lab julkaisi tutkimuksen, jossa tutkittiin DDoS-palveluita tarjoavia verkkosivuja [34]. Sivustot tarjoavat hyökkäyksiä, joissa hyökkäyksen hintaan vaikuttaa hyökkäyksen kesto ja sen koko. Esimerkiksi eräällä sivustolla viiden minuutin mittainen 125 Gbps hyökkäys maksoi viisi euroa. Jotkin DDoS-palvelut tarjoavat asiakkailleen erilaisia kuukausittain maksettavia tilauspaketteja, joissa asiakkaalla on rajattu määrä hyökkäysaikaa riippuen kuukausittain maksettavasta hinnasta. Maksaminen hoidetaan yleensä jollain kryptovaluutalla tai jonkin maksupalvelun kuten PayPalin välityksellä.

Kuvassa 1 on Bootyou-nimisen[1] palvelunestohyökkäyksiä myyvän verkkosivuston hinnasto. Verkkosivusto myy erilaisia paketteja, joissa vuokrataan DDoS-työkalun käyttöoikeus tietyn pituiseksi ajanjaksoksi. Vaihtoehtoina kyseisellä verkkosivustolla on 30 päivää tai elinikäinen käyttöoikeus. Muita paketin hintaan vaikuttavia seikkoja ovat: hyökkäyksen kaistanleveys, hyökkäysaika, samanaikaisten hyökkäysten määrä ja verkkosivuston tarjoaman tuen taso.

[1] Emme halua ohjata liikennettä rikollista toimintaa harjoittavalle sivustolle.



Plans - Select and choose from PayPal, Credit Card (Stripe), Bitcoin, Litecoin, Ethereum, Skrill or PaySafeCard & continue from there! (Instant delivery) | 25% off using Crypto!

Mobile Version

Name	Attack Time	Concurrent(s)	Plan Length	Attack Amount & Power	Server Access & Tools	Support	Price	Select
Bronze Monthly	300 Seconds	1 Concurrent	30 Days	Unlimited Attacks @ 10-15Gbps L4 and 5-10K R/s L7 Per Attack	Access To Tier 1 Servers & All Tools	Regular Support	$3	Purchase Now
Silver Monthly	700 Seconds	1 Concurrent	30 Days	Unlimited Attacks @ 10-15Gbps L4 and 5-10K R/s L7 Per Attack	Access To Tier 1 Servers & All Tools	Regular Support	$6	Purchase Now
Gold Monthly	1500 Seconds	1 Concurrent	30 Days	Unlimited Attacks @ 20-25Gbps L4 and 15-20K R/s L7 Per Attack	Access To Tier 2 Servers & All Tools	Premium Support	$12	Purchase Now
Platinum Monthly	3600 Seconds	1 Concurrent	30 Days	Unlimited Attacks @ 20-25Gbps L4 and 15-20K R/s L7 Per Attack	Access To Tier 2 Servers & All Tools	Premium Support	$20	Purchase Now
Extreme Monthly	7200 Seconds	2 Concurrents	30 Days	Unlimited Attacks @ 30-35Gbps L4 and 20-25K R/s L7 Per Attack	Access To Tier 3 Servers & All Tools	Premium Support	$35	Purchase Now
Ultra Monthly	10800 Seconds	3 Concurrents	30 Days	Unlimited Attacks @ 45-50Gbps L4 and 35-40K R/s L7 Per Attack	Access To Tier 4 Servers & All Tools	Premium Support	$40	Purchase Now
Bronze Lifetime	300 Seconds	1 Concurrent	Lifetime	Unlimited Attacks @ 10-15Gbps L4 and 5-10K R/s L7 Per Attack	Access To Tier 1 Servers & All Tools	Regular Support	$10	Purchase Now
Silver Lifetime	700 Seconds	1 Concurrent	Lifetime	Unlimited Attacks @ 10-15Gbps L4 and 5-10K R/s L7 Per Attack	Access To Tier 1 Servers & All Tools	Regular Support	$15	Purchase Now
Gold Lifetime	1500 Seconds	1 Concurrent	Lifetime	Unlimited Attacks @ 20-25Gbps L4 and 15-20K R/s L7 Per Attack	Access To Tier 2 Servers & All Tools	Premium Support	$20	Purchase Now
Platinum Lifetime	3600 Seconds	1 Concurrent	Lifetime	Unlimited Attacks @ 20-25Gbps L4 and 15-20K R/s L7 Per Attack	Access To Tier 2 Servers & All Tools	Premium Support	$35	Purchase Now
Extreme Lifetime 25% Off	7200 Seconds	2 Concurrents	Lifetime	Unlimited Attacks @ 30-35Gbps L4 and 20-25K R/s L7 Per Attack	Access To Tier 3 Servers & All Tools	Premium Support	$49.99	Purchase Now
Ultra Lifetime	10800 Seconds	3 Concurrents	Lifetime	Unlimited Attacks @ 45-50Gbps L4 and 35-40K R/s L7 Per Attack	Access To Tier 4 Servers & All Tools	Premium Support	$100	Purchase Now

Kuva 1. Bootyou sivuston DDoS-palvelun hinnasto 14. kesäkuuta 2018.

2.2. Bottiverkko

Bottiverkko koostuu joukosta internetiin yhdistyneitä saastuneita laitteita, joiden tietoturva on murrettu ja hallinta on siirtynyt kolmannelle osapuolelle. Bottiverkkoa käytetään yleensä palvelunestohyökkäyksiin tai roskapostin levittämiseen ja se voi koostua jopa sadoista tuhansista saastuneista laitteista. Bottiverkon toimintaa ohjataan C&C -järjestelyllä, jossa bottiverkon laitteet kommunikoivat C&C-palvelimen kanssa. C&C-järjestelyjä on olemassa kahdenlaisia: keskitettyjä ja hajautettuja. Keskitetyssä järjestelyssä bottiverkon omistaja antaa komentoja bottiverkolle C&C-palvelimen välityksellä. Kommunikaatio tyypillisesti tapahtuu IRC-protokollan (Internet Relay Chat) [35] välityksellä, missä bottiverkon omistaja pystyy antamaan komentoja reaaliajassa bottiverkolleen. McCarty [36] havainnoi jo vuonna 2003 ilmiötä, jossa saastuneet laitteet liittyivät bottiverkon omistajan IRC-kanavalle odottamaan käskyjä. Myös HTTP-protokollaan pohjautuvia C&C-järjestelyjä on, missä bottiverkon laitteet hakevat ohjeensa C&C-palvelimelta säännöllisin aikavälein [37]. Koska keskitetty C&C-järjestely voidaan torjua estämällä C&C-palvelimen IP-osoite, jotkin bottiverkot käyttävät hajautettua C&C-järjestelyä käyttäen vertaisverkkoa (eng. peer to peer)

apunaan [38]. Tämä hajautettu lähestymistapa tekee bottiverkon kaatamisesta vaikeampaa.

2.2.1. IoT-laitteen saastuttaminen osaksi bottiverkkoa

IoT-laitteen saastuttaminen osaksi bottiverkkoa tapahtuu tyypillisesti erilaisten haittaohjelmien avulla. Haittaohjelmat skannaavat internetiä löytääkseen potentiaalisia laitteita, joita saastuttaa. Yleensä haittaohjelmat käyttävät väsytyshyökkäystä (engl. brute-force attack) käyden läpi listan tunnetuista käyttäjätunnuksista ja salasanoista tai hyväksikäyttävät laitteessa esiintyvää tunnettua haavoittuvuutta. Mikäli loppukäyttäjä ei ole vaihtanut oletussalasanaa laitteeseen tai laitteesta löytyy jokin tunnettu haavoittuvuus, haittaohjelma voi saada laitteen hallintaansa. Haittaohjelman saatua laite hallintaansa laite ottaa yhteyttä palvelimelle, josta pahantahtoinen koodi ladataan laitteen muistiin.

2.2.2. Tunnettuja bottiverkkohaittaohjelmia

Bashlite/Lizkebab/Qbot/Torlus/LizardStresser on haittaohjelma, joka kohdistaa hyökkäyksensä Linux-järjestelmiin, joista löytyy avoin telnet-portti (portti 23 tai 2323). Haittaohjelma käyttää väsytyshyökkäystä hyväkseen laitteen kaappaamiseen. Bashliten ohjelmakoodissa on kovakoodattu C&C-palvelimen IP-osoite, joten haittaohjelman saastuttamia laitteita on helppo seurata. Level 3 Threat Research Labs teki tutkimuksen, joka osoitti että 96% Bashlite-haittaohjelman tekemistä bottiverkoista koostui IoT-laitteista, joista valtaosa oli kameroita tai digitaalisia videontallentimia [39]. Bashliten lähdekoodi julkaistiin vuonna 2015 ja siitä on sen jälkeen ilmestynyt useita erilaisia versioita. Bashlitea pidetään Mirain edeltäjänä.

Mirai-niminen haittaohjelma on ehkä tunnetuin IoT-laitteiden saastuttamiseen käytetty haittaohjelma. Wang ym. kuvaa Mirain toimintaa seuraavasti: Mirai kohdistaa hyökkäyksensä Linux-pohjaisiin IoT-laitteisiin, joissa on avoin telnet-portti, kokeilemalla väsytyshyökkäyksellä erilaisia oletus käyttäjätunnuksia ja salasanoja saadakseen IoT-laitteen hallintaansa [40]. Mirain murrettua laitteen suojauksen, se lataa haittaohjelmakoodin laitteen muistiin ja laitteesta tulee osa bottiverkkoa. Mirain luoman bottiverkon laitteet skannaavat aktiivisesti internetiä etsien uusia kohteita. Mirai-haittaohjelma pyrkii myös poistamaan muut kilpailevat haittaohjelmat saastuneesta laitteesta ja estämään muita haittaohjelmia kaappaamasta laitetta itselleen. Se varastoi itsensä saastuneen laitteen keskusmuistiin (RAM), joten yksinkertainen laitteen uudelleenkäynnistys pystyy poistamaan sen. Mirain lähdekoodi julkistettiin syyskuussa vuonna 2016, mikä on johtanut monien erilaisten varianttien ilmestymiseen. Akamai [11] arvioi että Mirai-bottiverkolla oli vuoden 2017 lopussa noin 150 000 eri IP-osoitetta hallussaan. Suurin määrä IP-osoitteita Mirai-bottiverkolla oli hallussa vuoden 2016 lopussa hyökkäyksessä Deutsche Telekomia vastaan [41], jolloin osoitteiden määrä oli liki kolme miljoonaa.

Radwaren tietoturvatiimi havaitsi huhtikuussa vuonna 2017 bottiverkkohaittaohjelman nimeltä Brickerbot [42]. Päinvastoin kuin muut bottiverkkohaittaohjelmat, brickerbotin tarkoitus on tuhota saastuneita tai haavoittuvia IoT-laitteita korruptoimalla niiden muisti. Brickerbot käyttää samaa hyökkäysvektoria kuin Mirai ja Bashlite etsien avoimia telnet-portteja. Brickerbotin kehittäjä janit0r julkaisi manifestin vuoden 2017 lopussa, missä hän kertoi “Internet Chemotherapy” -

projektistaan [43]. Manifestissa kerrotaan yksityiskohtaisesti motivaatiosta brickerbotin takana ja sen tehtävästä. Janit0r kertoo brickerbotin vähentäneen DDoS-hyökkäysten määrää tilapäisesti ja antaa lukijoilleen neuvoja, miten he voivat suojella internetiä bottiverkkojen kasvavalta uhalta.

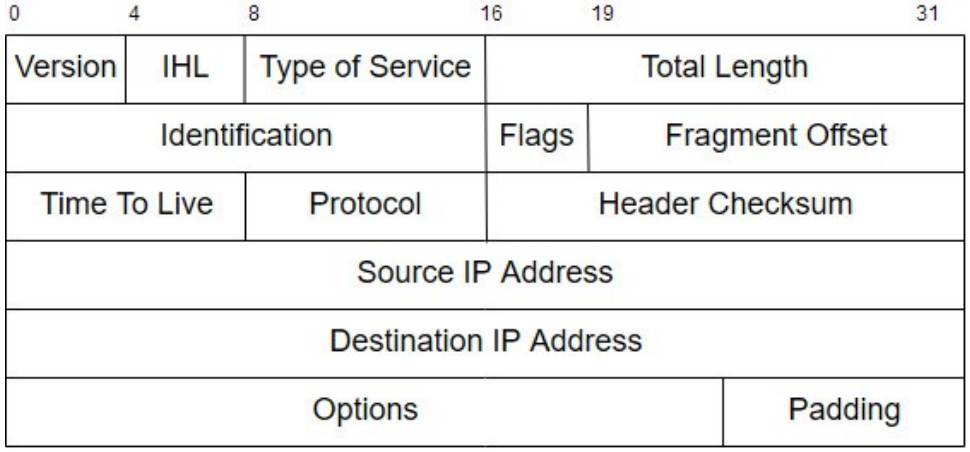
Viestintävirasto tiedotti marraskuussa 2017 Reaper-nimisestä haittaohjelmasta, joka saastuttaa IoT-laitteita [44]. Reaper on hieman Miraita edistyneempi haittaohjelma ja se hyväksikäyttää päivittämättömissä laitteissa olevia haavoittuvuuksia. Haavoittuviin laitteisiin kuuluu joukko erilaisia reitittimiä ja IP-kameroita. Vielä ei ole tiedossa, mihin Reaper-bottiverkkoa tullaan käyttämään, mutta saastuneiden laitteiden määrän perusteella uhka on suuri [45].

3. TIETOLIIKENNEPAKETIT JA NIIDEN TARKKAILEMINEN

Internetissä kommunikaatio tapahtuu tietoliikennepakettien välityksellä. Tietoliikennepaketit ovat yksiköitä, jotka sisältävät otsikon ja kuljetettavan tietosisällön. Tarkkailemalla tietoliikennepaketteja, voidaan niistä kerätä lukuisia tietoja, kuten vastaanottajan ja lähettäjän IP-osoitteet, käytetty protokolla ja jopa lähetetty tietosisältö, jos salausta ei ole käytetty. Pakettien tarkkaileminen mahdollistaa verkkoliikenteen tarkan analysoimisen ja valvomisen.

3.1. Tietoliikennepaketti

OSI-mallissa kolmannen kerroksen eli verkkokerroksen tiedonsiirtoyksiköitä kutsutaan tietoliikennepaketeiksi [46]. Tietoliikennepaketti on yksikkö, joka koostuu otsikkotiedoista ja tietosisällöstä. IP-paketin otsikko on esitetty kuvassa 2[1]. Otsikko löytyy jokaisesta IP-paketista ja sitä seuraa lähetetyn tietosisällön sisältävä kenttä. Tärkein otsikosta löytyvä tieto on paketin lähettäjän ja vastaanottajan IP-osoitteet, sekä käytetty protokolla. Käytössä olevia protokollia on lukuisia ja niiden myöntämisestä ja hallitsemisesta vastaa Internet Assigned Numbers Authority -instituutio [47]. Käytetyimpiin protokolliin kuuluvat TCP, UDP ja ICMP muodostavat valtaosan DDoS-hyökkäyksiin käytetyistä protokollista [21,22,23].



Kuva 2. IP-otsikko

3.1.1. TCP (Transmission control protocol)

TCP [48] on tietoliikenneprotokolla, jonka tarkoitus on tarjota hyvin luotettava yhteys kahden internetiin yhdistyneen laitteen välillä. TCP varmistaa, että paketit vastaanotetaan oikeassa järjestyksessä ja että virheelliset paketit lähetetään uudelleen.

TCP-yhteys kahden internetiin yhdistetyn laitteen välillä muodostetaan kolmivaiheisella kättelyllä, missä yhteyden aloittava osapuoli lähettää SYN-paketin, johon toinen osapuoli vastaa SYN/ACK-paketilla. Saatuaan SYN/ACK-paketin aloittava osapuoli vastaa toiselle osapuolelle ACK-paketilla, jonka jälkeen tietosisällön siirto voi alkaa.

TCP-yhteyden lopetus tapahtuu kolmi- tai nelivaiheisella kättelyllä. Nelivaiheisessa kättelyssä osapuoli A ilmoittaa haluavansa lopettaa yhteyden lähettämällä FIN-paketin. Osapuoli B vastaa FIN-pakettiin ACK-paketilla ja FIN-paketilla. Saatuaan FIN-paketin

Osapuoli A lähettää vielä ACK-paketin, jonka vastaanotettuaan osapuoli B sulkee yhteyden. Kolmivaiheisessa kättelyssä osapuoli B vastaa FIN/ACK paketilla, jonka saatuaan osapuoli A lähettää ACK-paketin. Viimeisen ACK-paketin vastaanotettua osapuoli B sulkee yhteyden.

3.1.3. UDP (User datagram protocol)

UDP [49] on tietoliikenneprotokolla, joka ei vaadi aktiivisen yhteyden muodostamista kättelyllä kuten TCP. UDP ei myöskään sisällä virheenhallintaa, mistä johtuen sitä voidaan käyttää tiedonsiirtoon, jossa virheiden merkitys on pieni tai korjaaminen on liian hidasta, tai jo liian myöhäistä (esim. reaaliaikaiset online moninpelit). Useat eri internetpalvelut, jotka tarvitsevat nopeaa yhteydetöntä tiedonsiirtoa käyttävät UDP:tä, kuten esimerkiksi DNS-palvelut, joiden tiedonsiirto koostuu lyhyistä kyselyistä, jotka vaativat nopeita vastauksia.

3.1.4. ICMP (Internet control message protocol)

ICMP [50] on tietoliikenneprotokolla, jonka tarkoitus on lähettää virheviestejä tai muuta, esimerkiksi palvelimien ja reitittimien toimintaan liittyvää informaatiota. ICMP-viestejä lähetetään esimerkiksi tilanteessa, jossa lähetetty paketti ei pystynyt saavuttamaan määränpäätään, tai kun reitittimellä ei ole kapasiteettia ohjata pakettia eteenpäin. ICMP-pakettien muunneltavaa kokoa on käytetty hyväksi “Ping of death” -hyökkäyksessä, jossa lähetetään suuren tietosisällön sisältäviä ping-paketteja [51].

3.2. Pakettien tarkkaileminen

Pakettien tarkkaileminen eli nuuskiminen tarkoittaa menetelmää, jossa luetaan tietoverkossa liikkuvien pakettien sisältöä. Pakettien tarkkailemisen tyypillisesti hoitaa ohjelma, jota ajetaan laitteessa, minkä läpi tietovirta kulkee. Paketit sisältävät tiedon lähettäjän ja vastaanottajan IP-osoitteista, käytetystä protokollasta, lipuista, portista ja tietosisällöstä. Näitä tietoja analysoimalla voidaan tarkasti valvoa verkon toimintaa [52].

Pakettien tarkkaileminen vaatii lähes aina järjestelmänvalvojan oikeudet, sillä kaikkien läpimenevien pakettien sisällöt ovat tarkkailijan nähtävissä. Mikäli käytetty protokolla ei sisällä minkäänlaista salausta, myös paketin tietosisältö on täysin tarkkailijan nähtävissä, mikä on vaarallista, jos tietosisältö sisältää jotain arkaluontoista informaatiota, kuten käyttäjätunnuksia ja salasanoja.

Pakettien tarkkaileminen on suuressa osassa erilaisia tietoliikenneverkkoihin kohdistuvia hyökkäyksiä. Esimerkiksi MITM-hyökkäyksessä (Man in the middle) tarkkailemisen tekevä laite saadaan kahden kommunikoivan laitteen väliin ilman että kumpikaan osapuoli huomaa sitä. Tämä voidaan saavuttaa lähiverkossa ARP cache poisoning -menetelmällä, jossa tarkkaileva osapuoli tekeytyy kommunikaation eri osapuoliksi väärentämällä oman MAC-osoitteensa kommunikoivien laitteiden ARP-tauluihin. [53]

4. TYÖN KUVAUS

Työssä tehty ohjelma on kirjoitettu Python-ohjelmointikielellä. Ohjelmaa suoritetaan Raspberry Pi 2 -tietokoneella, joka toimii langattoman verkon tukiasemana IoT-laitteille. Ohjelman tarkoitus on havaita DDoS-hyökkäykseen osallistuva IoT-laite, joka on yhdistyneenä tukiasemaan. Ohjelma soveltuu myös suoritettavaksi reitittimillä, joissa on Linux-pohjainen käyttöjärjestelmä ja tuki 4.1. luvussa listattuihin ohjelmiin.

Havaitsinlaitteena käytettyyn Raspberry Pi 2 -tietokoneeseen kohdistui työn aikana useita sanakirja- ja väsytyshyökkäyksiä, minkä vuoksi laitteen turvallisuutta parannettiin luomalla skripti, joka pudottaa SSH-yhteyden kymmenen väärän kirjautumisyrityksen jälkeen.

4.1. Käytetty laitteisto ja ohjelmisto

Työssä käytetään Raspberry Pi 2 model B tietokonetta, jonka käyttöjärjestelmänä toimii Raspbian 8.0 (jessie). Tietokoneessa on 1 Gigatavun keskusmuisti ja neliytiminen 900 MHz:n kellotaajuudella toimiva ARM Cortex-A7 -suoritin. Työssä käytetään lisäksi USB-porttiin kytkettävää WLAN-adapteria. Ohjelman toteuttamiseen käytettiin Python-ohjelmointikielen versiota 2.7.9 ja pakettien tarkkailemiseen ja manipulointiin tarkoitettua Scapy-nimisen pythonkirjaston versiota 2.3.3[1]. Lisäksi ohjelmassa käytettiin Pythonin time-, json- ja multiprocessing-kirjastoja. Pakettien tarkkaileminen Scapylla vaati myös tcpdump-ohjelman version 4.9.0[2]. Tukiaseman luomiseen käytettiin hostapd-paketin versiota 2.3[3]. Informaation esittämiseen verkkosivulla käytettiin Apache HTTP Server -palvelinohjelmaa[4] ja ohjelmalta saadun JSON-pohjaisen datan jäsentämiseen käytettiin PHP-skriptiä.

4.2. Havaitsinlaitteen turvallisuus

Työssä havaitsinlaitteena käytetty Raspberry Pi on myös IoT-laite, joka on altis ulkoverkosta tuleville hyökkäyksille. Työn aikana ohjelman parissa työskentelyn helpottamiseksi Raspberry Pi -tietokoneessa käytettiin SSH-palvelinta, joka mahdollisti laitteen etäkäytön. Laitteen SSH-portti joka oli auki ulkoverkkoon keräsi jo muutaman päivän aikana useita väsytys- ja sanakirjahyökkäyksiä, joilla yritettiin todennäköisesti kaapata laite osaksi bottiverkkoa. Kuvassa 3 on esitetty osa lokitiedostoa, jossa näkyy 193.201.244.218 IP-osoitteesta tuleva sanakirjahyökkäys, jossa hyökkääjä kokeilee eri käyttäjä/salasana-yhdistelmiä.

Hyökkäysten torjumiseksi luotiin skripti, joka pudottaa yhteyden, kun samasta IP-osoitteesta on yritetty kirjautua virheellisillä tunnuksilla sisään yli kymmenen kertaa peräkkäin. Lisäksi laitteelle luotiin jokaiselle käyttäjälle omat tunnukset omilla salasanoillaan ja oletuskäyttäjä poistettiin käytöstä.

[1] https://scapy.net/
[2] http://www.tcpdump.org/
[3] https://w1.fi/hostapd/
[4] https://httpd.apache.org/

```
sshd[25664]: Invalid user git from 193.201.224.218
sshd[25664]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=193
sshd[25664]: Failed password for invalid user git from 193.201.224.218 port 21157 ssh2
sshd[25672]: Invalid user halt from 193.201.224.218
sshd[25672]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=193
sshd[25672]: Failed password for invalid user halt from 193.201.224.218 port 62202 ssh2
sshd[25683]: Invalid user HELLO from 193.201.224.218
sshd[25683]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=193
sshd[25683]: Failed password for invalid user HELLO from 193.201.224.218 port 34186 ssh2
sshd[25692]: Invalid user helpdesk from 193.201.224.218
sshd[25692]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=193
sshd[25692]: Failed password for invalid user helpdesk from 193.201.224.218 port 4117 ssh2
```

Kuva 3. Leike Raspberry Pi:n SSH-palvelimen lokitiedostosta

4.3. Ohjelman suunnittelu

Ohjelman ensimmäinen versio tarkkaili paketteja sisäverkon ulkopuolelta, mutta se pystyi havaitsemaan ainoastaan broadcast- ja multicast-paketit. Tästä johtuen Raspberry Pi:stä tehtiin langattoman verkon tukiasema, johon IoT-laitteet yhdistyvät. Koska kaikki liikenne kulkee Raspberry Pi:n WLAN-sovittimen läpi, ohjelma pystyy tarkkailemaan ja analysoimaan sitä.

Nuuskimisen jälkeen paketit erotellaan eri protokollien mukaan. Ohjelma suunniteltiin havaitsemaan neljä eri DoS-hyökkäystyyppiä: DNS-tulva, UDP-tulva, TCP SYN -tulva, ICMP-tulva. Kyseiset neljä DoS-hyökkäystyyppiä valittiin, koska suurin osa lähiaikoina tehdyistä palvelunestohyökkäyksistä kuuluu näihin neljään protokollaan [21] ja Scapy-kirjasto tarjosi hyvät keinot erotella kyseisiin protokolliin kuuluvat paketit.

Ohjelman käyttämät DoS-hyökkäystyyppien havaintomenetelmät:

- DNS-tulva havaitaan, mikäli yhteen IP-osoitteeseen on mennyt kynnysarvoa enemmän paketteja viimeisen 30:en sekunnin aikana.

- TCP SYN -tulvan havaitsemiseen käytetään pakettien lukumäärän seuraamisen lisäksi lisäehtona FIN ACK- ja FIN-pakettien suhdetta lähetettyihin SYN-paketteihin. Ehto täyttyy, mikäli suhde on lähellä nollaa. Normaalissa TCP-kommunikaatiossa SYN-pakettien ja FIN tai FIN ACK -pakettien suhteen tulisi olla yksi tai lähellä sitä [54].

- UDP-tulvan havaitsemiseen katsotaan lähetettyjen pakettien lukumäärä yhteen IP-osoitteeseen. Ohjelma ilmoittaa palvelunestohyökkäyksestä, mikäli kynnysarvo ylittyy.

- ICMP-tulvan havaitsemiseen käytetään samaa menetelmää kuin UDP-tulvan havaitsemiseen.

Ohjelman suoritus osoittautui ongelmalliseksi, sillä verkkoliikenteen tarkkailun ja analysoinnin pitää tapahtua samanaikaisesti, jotta kaikki verkkoliikenne saataisiin tutkittua. Ongelma johtui Python-kääntäjän säieturvallisesta ominaisuudesta nimeltä GIL (Global Interpreter Lock). GIL antaa ainoastaan yhden säikeen suorittaa pythonkomentoja kerrallaan. Ongelma ratkaistiin käyttämällä pythonin multiprocessing

-moduulia, joka mahdollistaa useamman prosessin ajamisen samanaikaisesti, jolloin ohjelma suorittaa kahta identtistä prosessia, jotka oikein ajoitettuna tarkkailevat ja analysoivat verkkoliikennettä vuorotellen.

4.4. Langattoman tukiaseman luominen

Tukiaseman luomiseksi Raspberry Pi:hin asennettiin ensin DHCP-palvelin ja se asetettiin jakamaan kytketyille laitteille automaattisesti verkkoasetukset 192.168.2.0/24 IP-avaruuteen, jonka jälkeen laitteeseen asennettiin hostapd-ohjelma, joka mahdollistaa laitteen toimimisen langattomana tukiasemana. Lisäksi Raspberry Pi:n IP -osoite määritettin staattiseksi. Raspberry Pi:ssa otettiin myös käyttöön osoitteenmuunnos (NAT) sysctl.conf -asetustiedostossa. Sen jälkeen tehtiin kuvassa 4. näkyvät iptables POSTROUTING ACCEPT ja OUTPUT ACCEPT -säännöt.

```
# Generated by iptables-save v1.4.21 on Thu Mar  8 14:53:45 2018
*nat
:PREROUTING ACCEPT [4:627]
:INPUT ACCEPT [4:627]
:OUTPUT ACCEPT [1:76]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Thu Mar  8 14:53:45 2018
# Generated by iptables-save v1.4.21 on Thu Mar  8 14:53:45 2018
*filter
:INPUT ACCEPT [581:54228]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [314:39891]
-A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i wlan0 -o eth0 -j ACCEPT
COMMIT
# Completed on Thu Mar  8 14:53:45 2018
```

Kuva 4. Iptables-säännöt

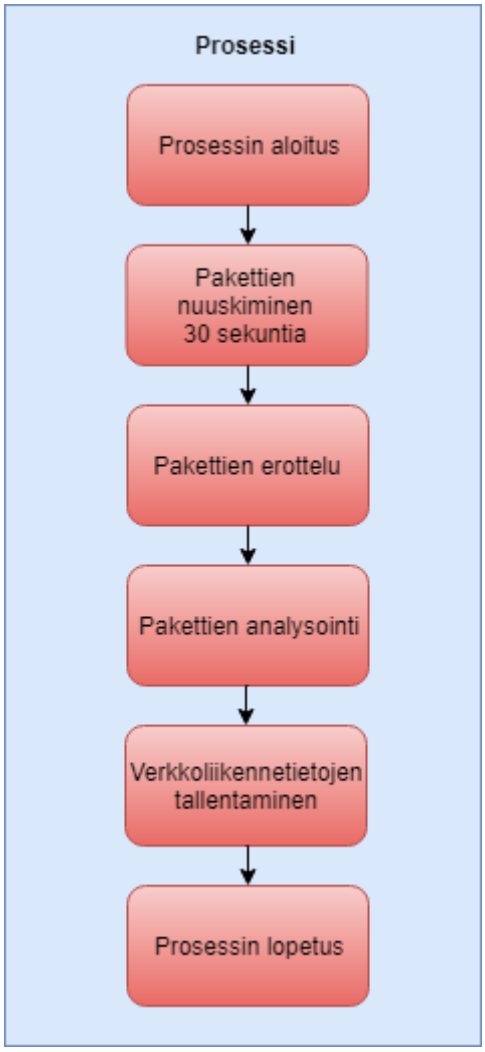
4.5. Ohjelman kuvaus

Ohjelma tarkkailee tukiasemaan yhdistyneiden IoT-laitteiden verkkoliikennettä ja pyrkii havaitsemaan, mikäli jokin yhdistyneistä IoT-laitteista osallistuu palvelunestohyökkäykseen.

Tukiasemaan yhdistyneiden IoT-laitteiden verkkoliikennettä tarkkaillaan 30:n sekunnin ajan, jonka jälkeen ohjelma luo uuden prosessin, joka alkaa tarkkailuvaiheesta. Samanaikaisesti edellinen tarkkailun suorittanut prosessi jatkaa erotteluvaiheeseen. Kyseinen prosessien aikataulutus takaa, että verkkoliikennettä tarkkaillaan koko ajan ohjelman suorituksen aikana. Erotteluvaiheessa eri pakettityypit erotellaan protokollien mukaan omiin pakettilistoihinsa, jotka toimivat syötteinä eri analysointifunktiolle. Analysointivaiheessa tutkitaan tarkkailuvaiheessa nuuskittujen pakettien sisältö, jonka perusteella arvioidaan, osallistuuko jokin laitteista palvelunestohyökkäykseen. Kuva 5 havainnollistaa yksittäisen ohjelman luoman prosessin toimintaa.

Mikäli jonkin laitteen epäillään osallistuneen palvelunestohyökkäykseen, ohjelma tallentaa kyseisen laitteen tiedot paikalliseen tiedostoon JSON-muotoisena, jonka

jälkeen prosessi lopettaa itsensä. JSON sisältää hyökkäykseen osallistuvan laitteen IP- ja MAC-osoitteen, hyökkäyksen tuntomerkit täyttävien pakettien määrän ja hyökkäystyyppiin liittyvät lisätiedot. Ohjelman tallentama hyökkäykseen liittyvä informaatio on luettavissa paikalliselta verkkosivulta, joka hakee ja jäsentää PHP -skriptin avulla tallennetussa tiedostossa olevan JSON-muotoisen datan.



Kuva 5. Yksittäisen prosessin toiminta

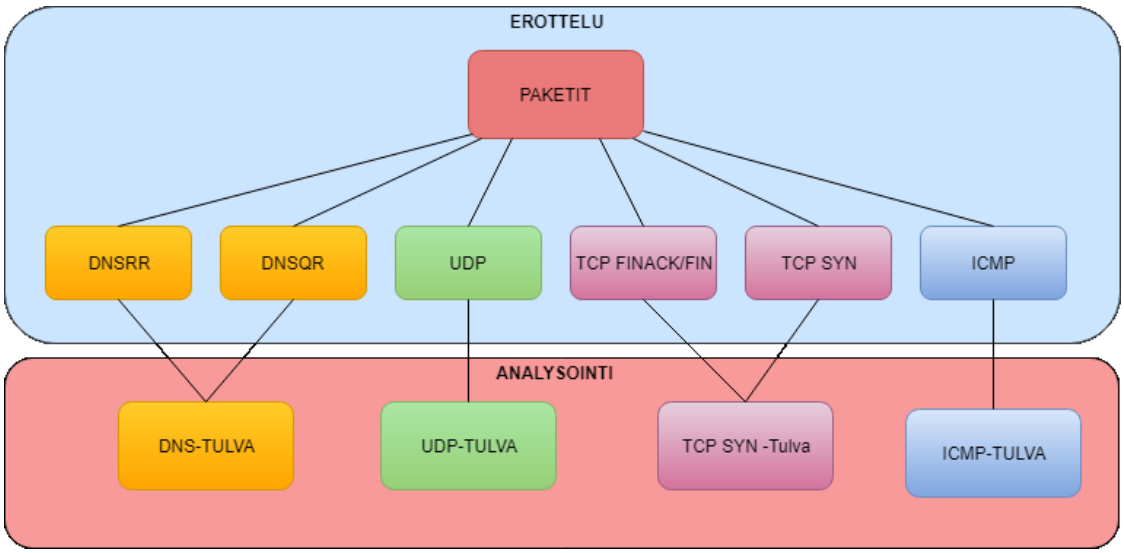
4.5.1. Verkkoliikenteen kuuntelu

Ohjelma käyttää pakettien tarkkailemiseen Scapy-kirjastosta löytyvää sniff-funktiota. Sniff-funktio ottaa argumenteikseen Berkeley packet filter[1] -muotoa olevan suodatin-merkkijonon, nuuskimiseen käytettävän sovittimen nimen ja nuuskimisen keston. Sniff-funktio palauttaa listan nuuskituista paketeista, joka suodatetaan annetulla suodatin-merkkijonolla "*tcp or udp or icmp*". Suodatin hylkää kaikki muut paketit, jotka eivät kuulu kyseisiin protokolliin. Sovitin, jota ohjelma tarkkailee, on Raspberry Pi:n WLAN-sovitin, johon IoT-laitteet ovat yhdistetty. Tarkkailun kestoksi on asetettu 30 sekuntia.

[1] http://biot.com/capstats/bpf.html

4.5.2. Pakettien erittely ja analysointi

Ohjelma jakaa tarkkailuvaiheessa nuuskitut paketit kuuteen eri ryhmään: UDP-paketit, TCP SYN -paketit, TCP FINACK/FIN -paketit, DNSRR-paketit, DNSQR-paketit ja ICMP-paketit. Pakettien jaottelun jälkeen paketit analysoidaan tyypeittäin ja pyritään havaitsemaan neljä eri palvelunestohyökkäystyyppiä: DNS-tulva, UDP-tulva, TCP SYN -tulva ja ICMP-tulva. Kuva 6 esittää mitä paketteja käytetään eri DDoS-hyökkäystyyppien havaitsemiseen.



Kuva 6. Pakettien erottelu analysointivaiheeseen.

DNS-tulvan havaitseminen

DNS-tulvan havaitsemiseksi ohjelma muodostaa molemmista DNS Resolve- ja DNS Query -pakettilistoista sanakirjat, joista nähdään pakettien määrä eri IP-osoiteparien välillä. Jos lähetettyjen Query-pakettien määrä ylittää kynnysarvon tai Query-pakettien lähettäjän IP-osoitetta ei löydy Resolve-pakettien vastaanottajien IP-osoitteista, voidaan päätellä, että kyseessä on palvelunestohyökkäys. Havaittuaan hyökkäyksen ohjelma tallentaa laitteen IP- ja MAC-osoitteen ja lähetettyjen pakettien määrän.

UDP-tulvan havaitseminen

UDP-tulvan havaitsemiseksi ohjelma käy läpi muista paketeista erotellut UDP-paketit ja muodostaa paketeista löytyvistä IP-osoitepareista sanakirjan, jossa avaimena toimii IP-osoitepari ja arvona pakettien määrä näiden kahden IP-osoitteen välillä. UDP-tulvan tapauksessa ainoastaan ulospäin menevät paketit lasketaan. Sanakirjan IP-osoitepari, jonka arvo on suurin, käydään läpi tarkemmin. Pakettimäärä näiden kahden IP-osoitteen välillä otetaan talteen, jonka jälkeen sitä verrataan asetettuun kynnysarvoon. Kynnysarvon ylittyessä ohjelma ilmoittaa palvelunestohyökkäyksestä tallentamalla tiedot palvelunestohyökkäykseen osallistuvan laitteen sekä IP- ja MAC-osoitteesta, että lähetetystä pakettimäärästä.

TCP SYN -tulvan havaitseminen

TCP SYN -tulvan havaitsemiseksi ohjelma käy läpi TCP SYN -paketit ja muodostaa paketeista löytyvistä IP-osoitepareista sanakirjan samalla periaatteella kuin UDP-tulvan havaitsemisessa. Seuraavaksi ohjelma tutkii tarkemmin IP-osoiteparin, joiden välillä on lähetetty eniten TCP SYN -paketteja. Koska ohjelman tarkoitus on havaita laite, joka osallistuu palvelunestohyökkäykseen, tarkistetaan myös, että TCP SYN -paketit ovat ulospäin menevää verkkoliikennettä.

Ohjelman löydettyä IP-osoiteparin, joiden välillä on lähetetty eniten TCP SYN -paketteja, se käy läpi kohde-IP-osoitteesta vastaanotetut TCP FIN ACK- ja TCP FIN -paketit, joiden lukumäärää verrataan lähetettyihin TCP SYN-paketteihin alla olevalla laskukaavalla 1.

$$\frac{TCP\ FIN\ ACK + TCP\ FIN}{TCP\ SYN} \quad (1)$$

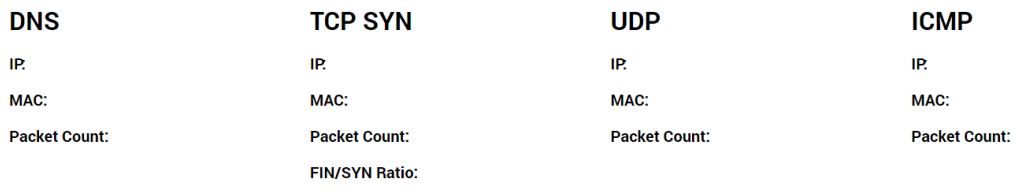
Ohjelma ilmoittaa palvelunestohyökkäyksestä, jos suhde on lähellä nollaa ja lähetettyjen TCP SYN -pakettien lukumäärä ylittää asetetun kynnysarvon. Tieto palvelunestohyökkäykseen osallistuvan laitteen IP- ja MAC-osoitteesta, lähetettyjen TCP SYN -pakettien lukumäärästä, sekä FIN ACK ja FIN -pakettien suhteesta TCP SYN -paketteihin tallennetaan.

ICMP-tulvan havaitseminen

ICMP-tulvan havaitsemiseksi käytetään samanlaista menetelmää kuin UDP-tulvan havaitsemiseksi.

4.5.3. Verkkosivu

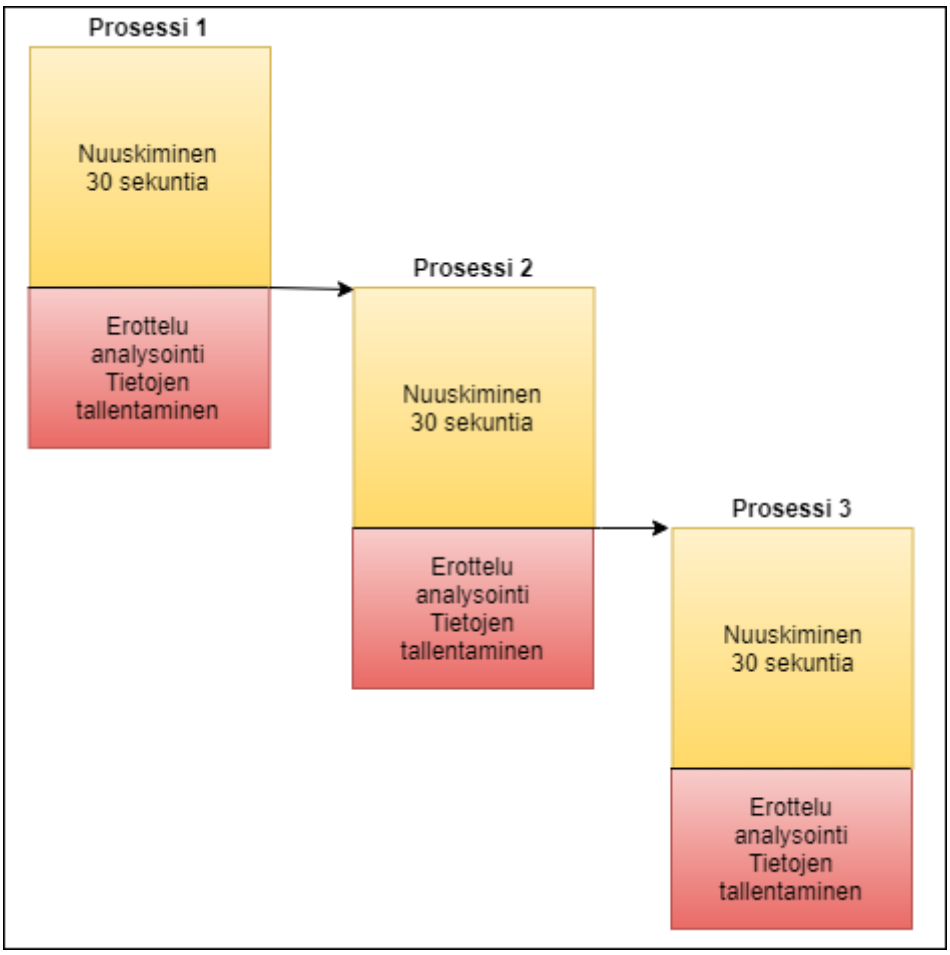
Käyttäjälle ilmoitetaan palvelunestohyökkäykseen osallistuvasta IoT-laitteesta verkkosivun avulla. Se hakee Python ohjelman luoman JSON-muotoisen datan PHP -skriptin avulla ja näyttää sen käyttäjälle ymmärrettävässä muodossa. Ohjelma päivittää tiedot aina kun prosessi on viimeistellyt analyysin, eli noin 30 sekunnin välein. Verkkosivulta käyttäjä näkee hyökkäävän laitteen IP-osoitteen, MAC-osoitteen, viimeisen 30 sekunnin aikana lähetettyjen pakettien määrän ja hyökkäystyypin analyysiin liittyvää lisätietoa. Verkkosivu ja sen näyttämät tiedot ovat esitetty kuvassa 7.



Kuva 7. Verkkosivu ja sen näyttämät tiedot.

4.5.4. Ohjelman multiprosessointi

Ohjelman käynnistyessä luodaan Pythonin multiprocessing-kirjastoa käyttäen yksi prosessi, joka aloittaa pakettien tarkkailun. Jotta kaikki paketit saadaan analysoitua ohjelma odottaa 30 sekuntia ja aloittaa uuden prosessin, jonka jälkeen aikaisempi prosessi siirtyy analysoimaan nuuskittuja paketteja. Yksittäinen prosessi sulkeutuu, kun analysointivaihe on suoritettu ja tiedot mahdollisesta palvelunestohyökkäyksestä tallennettu. Kuva 8 esittää miten ohjelman prosessit ajoittuvat suhteessa toisiinsa.



Kuva 8. Prosessien ajoitus.

5. OHJELMAN TESTAUS

Ohjelman kykyä havaita palvelunestohyökkäyksiä testattiin erilaisilla itsetehdyillä palvelunestohyökkäystä simuloivilla työkaluilla. Työkaluilla simuloitiin neljää eri DoS-hyökkäystyyppiä: DNS-tulvaa, TCP SYN -tulvaa, UDP-tulvaa ja ICMP-tulvaa. Hyökkäykset tehtiin tukiasemaan langattomasti yhdistetyltä Raspberry Pi 2 -tietokoneelta. Ohjelman toimintaa seurattiin paikalliselta verkkosivulta, joka ilmoitti, jos ohjelma havaitsi palvelunestohyökkäyksen viimeisen 30 sekunnin aikana. Ohjelmaa ajavan Raspberry Pi 2 -tietokoneen suoritinkäyttöä ja testeissä käytetyn verkkoliikenteen kaistanleveyttä mitattiin myös hyökkäysten aikana. Normaalia verkkoliikennettä simuloitiin suoratoistamalla videokuvaa VLC media playerillä toiselta Raspberry Pi:lta, sekä Plex Media Serverillä, jonka kautta suoratoistettiin videokuvaa ulkoverkossa olevalle laitteelle.

5.1. Testauksessa käytetty verkkoliikenne

Testauksessa simuloitiin DoS-verkkoliikennettä ja normaalia verkkoliikennettä. DoS-verkkoliikenteen simulointi toteutettiin itse tehdyillä DNS-, TCP SYN-, UDP- ja ICMP-tulva-työkaluilla. Normaalia liikennettä simuloitiin VLC media playerillä suoratoistamalla videokuvaa toiselta Raspberry Pi:lta toiselle, sekä Plex Media Server ohjelmalla, jonka avulla suoratoistettiin videokuvaa verkon ulkopuoliselle laitteelle. Simuloinnissa käytetyt pakettikoot ovat esitetty taulukossa 1.

UDP-, TCP SYN- ja ICMP-tulva-työkalut toteutettiin hyödyntäen pythonin multiprocessing- ja socket-kirjastoja. Jokaiseen työkaluun ohjelmoitiin luokka, joka avaa yhden pistokkeen pakettien lähettämistä varten. Luokkaa instantioimalla multiprocessing-kirjaston avulla saadaan aikaiseksi useampi prosessi, joista jokainen lähettää ennalta määritettyjä paketteja samaan osoitteeseen. UDP- ja ICMP-tulva-työkalujen pakettien tietosisältönä käytettiin satunnaisesti luotua bittijonoa, jonka avulla työkalun lähettämien pakettien kooksi saatiin kyseisten protokollien 1500 tavun maksimipituutta lähellä oleva 1492 tavun pakettikoko. UDP- ja ICMP-tulva-työkaluja testattiin myös tehokkaalla pöytätietokoneella sisäisen verkkoliittymän kautta, jolloin saavutettiin jopa 6 Gb/s kaistanleveys. Raspberry Pi:n verkkosovitin ei kyennyt testeissä kuin maksimissaan 30,10 Mb/s kaistanleveyteen, mistä voidaan päätellä, että työkalujen suorituskyky oli testeihin riittävä.

TCP SYN-tulva-työkalun suorituskyky kaistanleveydellisesti on huomattavasti pienempi kuin UDP- ja ICMP-työkalujen johtuen SYN-pakettien pienestä 60 tavun koosta. Siitä huolimatta TCP SYN-tulva-työkalu saavutti pöytätietokoneella 190 Mb/s, huomattavasti suuremmalla pakettimäärällä kuin UDP- ja ICMP-työkalut.

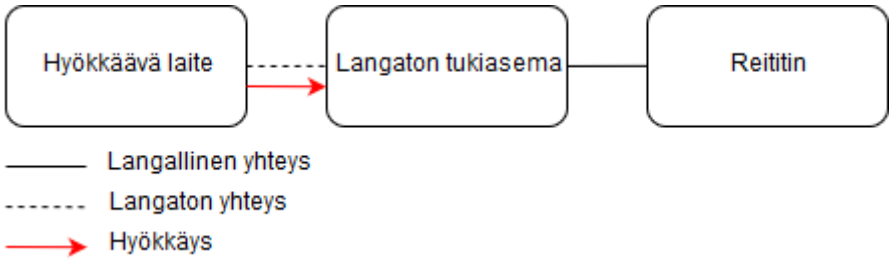
DNS-tulva-työkalu on toteutettu hyödyntäen Scapy-kirjaston paketteja lähettävää send-funktiota. UDP- TCP SYN- ja ICMP-tulva-työkalujen suorituskyky oli huomattavasti parempi DNS-tulva-työkaluun verrattuna, koska niissä matalan tason verkko-ohjelmointi oli hoidettu itse.

Taulukko 1. Verkkoliikenteen simuloinnissa käytetyt pakettikoot.

Simuloitu liikenne	Pakettikoko
DNS-tulva	73 tavua
TCP-SYN -tulva	60 tavua
UDP-tulva	1492 tavua
ICMP-tulva	1492 tavua
VLC Media Player	1370 tavua
Plex Media Server	Vaihteleva

5.2 Testausympäristö

Testausympäristö koostui kahdesta Raspberry Pi 2 -tietokoneesta ja internetiin yhdistyneestä reitittimestä. Ensimmäinen Raspberry Pi toimi langattomana tukiasemana, johon toinen Raspberry Pi oli yhdistettynä. Ohjelmaa suoritettiin ensimmäisessä Raspberry Pi:ssa ja testauksessa käytettävää verkkoliikennettä simuloitiin toisella Raspberry Pi:lla. Palvelunestohyökkäyksen kohde testeissä, joissa käytettiin DoS-työkalua, oli 192.168.2.1 IP-osoitteessa sijaitseva tukiasemana toimiva Raspberry Pi. Testausympäristö on esitetty kuvassa 9.



Kuva 9. Testausympäristö

5.3. Testaussuunnitelma

Testaus koostuu viidestä eri testistä:

- Testi 1: DNS-tulva-hyökkäys
- Testi 2: TCP SYN -tulva-hyökkäys
- Testi 3: UDP-tulva-hyökkäys
- Testi 4: ICMP-tulva-hyökkäys
- Testi 5: Normaali verkkoliikenne (VLC media player)
- Testi 6: Normaali verkkoliikenne (Plex Media Server)

Yksittäisen testin toteutus:

Ohjelma käynnistettiin tukiasemana toimivassa Raspberry Pi 2 -tietokoneessa, minkä jälkeen hyökkäyksen tekevässä Raspberry Pi:ssä käynnistettiin samanaikaisesti DoS-työkalu ja kaistanleveyttä mittaava nload-ohjelma. DoS-hyökkäyksen aikana ohjelman annettiin tehdä kaksi suorituskertaa, joista jälkimmäiseltä suorituskerralta mitattiin testeissä mitattavat asiat. Jokainen testi toistettiin viisi kertaa.

Testeissä mitatut asiat:

- Tunnistiko ohjelma palvelunestohyökkäyksen
  - Hyväksytyksi tunnistamiseksi lasketaan tilanne, jossa ohjelma tallentaa tiedot havaitusta hyökkäyksestä paikalliselle verkkosivulle.

- Ohjelman suoritinkäyttö
  - Ohjelman suoritinkäyttöä mitattiin testien aikana käyttämällä linuxin top-ohjelmaa ja tallentamalla sen antama suorittimen käyttöaste, jota ohjelma käytti.

- Palvelunestohyökkäyksen keskimääräinen kaistanleveys
  - Palvelunestohyökkäyksen keskimääräinen kaistanleveys mitattiin nload-ohjelmalla ohjelmaa ajavan Raspberry Pi 2 -tietokoneen WLAN-sovittimesta.

- Ohjelman verkkosivulle tallentamat tiedot hyökkäyksestä
  - Tiedot luettiin verkkosivulta.

- Palvelunestohyökkäyksessä lähetettyjen pakettien todellinen määrä
  - Lähetettyjen pakettien todellinen määrä laskettiin kaavalla 2 jakamalla hyökkäysliikenteen keskimääräinen kaistanleveys hyökkäyksessä käytettävien pakettien koolla. Tämän jälkeen arvo kerrottiin 30:llä, joka on yhden tarkkailujakson aika sekunteina.

$$pakettien\ määrä\ = \frac{bit/s}{paketin\ koko\ bitteinä} \times 30 \quad (2)$$

Testauksessa käytetyt ohjelman kynnysarvot ja lisäehdot palvelunestohyökkäyksen havaitsemiseksi ovat esitetty taulukossa 2.

Taulukko 2. Ohjelmassa testien aikana käytetyt kynnysarvot.

DoS-hyökkäystyyppi	**Pakettia/30 sekuntia**	**Lisäehto**
DNS-tulva	2500	IP ei ole vastaanotetuissa Resolve -paketeissa
TCP SYN -tulva	2500	F+FA/S -suhde alle 0.3
UDP-tulva	2500	-
ICMP-tulva	2500	-

5.4. Testien tulokset

Ohjelmaa testattiin kuudella eri testillä, joista neljässä testattiin ohjelman kykyä havaita palvelunestohyökkäyksiä eri DoS-työkaluilla. Viidennessä ja kuudennessa testissä ohjelmaa testattiin simuloimalla normaalia verkkoliikennettä suoratoistamalla videokuvaa tukiasemaan yhdistyneeltä laitteelta kahdella eri ohjelmalla. Testien aikana todettiin, että ohjelma saa kiinni vain pienen osan läpikulkevasta verkkoliikenteestä, mutta kykeni silti havaitsemaan kaikki palvelunestohyökkäykset sopivilla kynnysarvoilla. Testeissä 1-5 lähetettyjen pakettien pakettikoko tiedettiin, mikä mahdollisti todellisen lähetettyjen pakettien määrän laskemisen, jonka avulla saatiin selville, kuinka suuri osa paketeista jää ohjelmalta huomaamatta. Kaikki testit toistettiin viisi kertaa. Testeissä mitatut asiat ovat esitetty luvussa 5.3.

5.4.1. Testi 1 (DNS-tulva)

DNS-tulva-testin tulokset on esitetty taulukossa 3. Ohjelma tunnisti jokaisella testikerralla hyökkäyksen. Prosessorin suoritinkäyttö ohjelmaa ajettaessa pysyi tasaisesti 50% läheisyydessä, mikä tarkoittaa, että kaksi prosessoriydintä työskenteli lähes koko ajan täydellä kapasiteetilla. DNS-tulva-työkalun lähettämän hyökkäyksen kaistanleveys oli huomattavasti pienempi kuin muiden työkalujen lähettämien hyökkäysten, johtuen Scapy-kirjaston send-funktion huonosta suorituskyvystä ja pakettien pienestä koosta. Ohjelma sai lähes kaikki lähetetyt paketit kiinni, mikä nähdään vertaamalla verkkosivun ilmoittamaan pakettien määrää ja laskettua lähetettyjen pakettien määrää.

Taulukko 3. DNS-tulva-testin tulokset.

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Kyllä	46,3 %	88 kbit/s	IP: 192.168.4.18 MAC: 78:32:1b:91:8c:0e Packet Count: 4265	4520
2	Kyllä	49,9 %	85 kbit/s	IP: 192.168.4.18 MAC: 78:32:1b:91:8c:0e Packet Count: 4187	4366
3	Kyllä	49,3 %	87kbit/s	IP: 192.168.4.18 MAC: 78:32:1b:91:8c:0e Packet Count: 4280	4469
4	Kyllä	49,7 %	88kbit/s	IP: 192.168.4.18 MAC: 78:32:1b:91:8c:0e Packet Count: 4182	4520
5	Kyllä	49,3 %	88kbit/s	IP: 192.168.4.18 MAC: 78:32:1b:91:8c:0e Packet Count: 4209	4520

5.4.2. Testi 2 (TCP SYN-tulva)

TCP SYN -tulva-testin tulokset ovat esitetty taulukossa 4. Ohjelma tunnisti hyökkäyksen jokaisella testikerralla. TCP SYN-tulva-työkalun lähettämän hyökkäyksen kaistanleveys oli huomattavan suuri, ottaen huomioon, että lähetettyjen SYN-pakettien pakettikoko oli vain 60 tavua. Ohjelma sai kiinni vain noin 1,6% todellisesta lähetettyjen pakettien määrästä.

Taulukko 4. TCP SYN -tulva-testin tulokset.

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Kyllä	49,8 %	5,24 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4632 FIN/SYN Ratio: 0	327500
2	Kyllä	49,3 %	5,13 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5601 FIN/SYN Ratio: 0	320625
3	Kyllä	49,8 %	6,10 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4836 FIN/SYN Ratio: 0	381250
4	Kyllä	49,5 %	5,01 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5588 FIN/SYN Ratio: 0	313125
5	Kyllä	49,6 %	5,71 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5623 FIN/SYN Ratio: 0	356875

5.4.3. Testi 3 (UDP-tulva)

UDP-tulva-testin tulokset ovat esitetty taulukossa 5. Ohjelma tunnisti hyökkäyksen jokaisella testikerralla. UDP-tulva-hyökkäyksen lähettämä kaistanleveys oli huomattavasti suurempi kuin TCP SYN -tulvalla, johtuen UDP-tulvan käyttämästä suuremmasta 1492 tavun pakettikoosta. Kaistanleveys nousi neljän ensimmäisen testin aikana huomattavasti, mikä voi johtua erilaisista langattoman verkkoyhteyden häiriöistä ensimmäisillä testikerroilla. Ohjelma sai kiinni vain noin 9% lähetetyistä paketeista.

Taulukko 5. UDP-tulva-testin tulokset.

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Kyllä	49,0 %	15,19 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4905	38178
2	Kyllä	49,2 %	17,02 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 3575	42778
3	Kyllä	49,6 %	20,29 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5158	50996
4	Kyllä	49,8 %	28,40 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5136	71380
5	Kyllä	49,9 %	28,32 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 5207	71179

5.4.4. Testi 4 (ICMP-tulva)

ICMP-tulva-testin tulokset ovat esitetty taulukossa 6. Ohjelma tunnisti hyökkäyksen jokaisella testikerralla. ICMP-tulva-hyökkäys käytti 1492 tavun pakettikokoa ja oli kaistanleveydeltään suurin testauksessa käytetyistä hyökkäyksistä. Ohjelma sai kiinni noin 6% lähetetyistä paketeista.

Taulukko 6. ICMP-tulva-testin tulokset.

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Kyllä	50,0 %	29,92 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4377	75201
2	Kyllä	49,8 %	30,10 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4574	75427
3	Kyllä	49,4 %	29,98 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4644	75351
4	Kyllä	50,0 %	30,10 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4772	75427
5	Kyllä	50,0 %	30,02 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4193	75452

5.4.5. Testi 5 (Normaali verkkoliikenne: VLC media player)

Tulokset simuloimalla normaalia verkkoliikennettä käyttäen VLC media player -ohjelmaa ovat esitetty taulukossa 7. Ohjelma tunnisti VLC media playerin käyttämän verkkoliikenteen palvelunestohyökkäykseksi jokaisella testikerralla. VLC media playerin käyttämä verkkoliikenne koostuu 1370 tavun kokoisista UDP-paketeista, joten ohjelma luuli liikennettä UDP-tulva-hyökkäykseksi. Ohjelma sai ajoittain lähes kaikki lähetetyt paketit kiinni.

Taulukko 7. Testi 5:en tulokset (Normaali verkkoliikenne: VLC media player).

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Kyllä	43,9 %	1,39 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 3458	3804
2	Kyllä	47,9 %	2,14 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 3687	5857
3	Kyllä	47,5 %	2,40 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 4117	6569
4	Kyllä	44,9 %	1,40 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 3570	3832
5	Kyllä	42,5 %	1,45 Mbit/s	IP: 192.168.2.20 MAC: 78:32:1b:91:8c:0e Packet Count: 3427	3968

5.4.6. Testi 6 (Normaali verkkoliikenne: Plex Media Server)

Ohjelma ei tunnistanut Plex Media Serverin avulla tuotettua verkkoliikennettä hyökkäykseksi yhdelläkään testikerralla. Testin tulokset ovat esitetty taulukossa 8. Matala suoritinkäyttö johtuu ohjelman analyser-funktion vähäisestä käytöstä, sillä verkkoliikenne ei sovi minkään ohjelmassa tutkittavan DoS-hyökkäystyypin alle. Plex Media Server -ohjelman käyttämä verkkoliikenne koostuu lyhyistä TCP-yhteyksistä.

Taulukko 8. Testi 6:en tulokset (Normaali verkkoliikenne: Plex Media Server).

Testi	Tunnisti	Suoritinkäyttö	Kaistanleveys	Verkkosivun antamat tiedot	Laskettu lähetettyjen pakettien määrä
1	Ei	13,3 %	0,9 Mbit/s	-	-
2	Ei	32,6 %	2,25 Mbit/s	-	-
3	Ei	35,9,%	2,72 Mbit/s	-	-
4	Ei	24,9 %	2,52 Mbit/s	-	-
5	Ei	19,4 %	1,07 Mbit/s	-	-

6. POHDINTA

Ohjelma onnistui tunnistamaan testauksessa jokaisen DoS-työkaluilla tehdyn hyökkäyksen, mutta luuli myös VLC media playerin aiheuttamaa verkkoliikennettä palvelunestohyökkäykseksi. Pakettien tarkkailun hoitavan Scapy-kirjaston sniff-funktion huono suorituskyky suuren kuormituksen alla johti siihen, että suuri osa läpi kulkevista paketeista jäi havaitsematta. Testauksessa käytettyjen itsetehtyjen DoS-työkalujen suorituskyky oli riittävä, sillä suurin osa niistä saavutti ohjelman maksimikapasiteetin pakettien tarkkailuun. Ainoastaan DNS-tulva-hyökkäyksestä saatiin lähes kaikki paketit havaittua.

Ohjelman käyttämiä kynnysarvoja eri palvelunestohyökkäysten havaitsemiseen voitaisiin testien perusteella nostaa, mikä vähentäisi tapauksia, joissa normaali verkkoliikenne tunnistetaan palvelunestohyökkäykseksi. Jatkokehityksen kannalta olennaisinta olisi ohjelman pakettien tarkkailukyvyn nostaminen, minkä avulla voitaisiin asettaa tarkemmat kynnysarvot ja ohjelman kyky havaita palvelunestohyökkäyksiä parantuisi.

6.1. DDoS-hyökkäyksen tunnistusheuristiikka ja kynnysarvot

Normaalin liikenteen testissä ohjelma tunnisti VLC playerin lähettämän verkkoliikenteen palvelunestohyökkäykseksi. Tämä johtuu ohjelman UDP-tulvan tunnistusheuristiikasta, sillä kynnysarvona hyökkäyksen havaitsemiselle ohjelmassa käytettiin suhteellisen alhaista 2500 paketin pakettimäärää 30 sekunnin aikana. Mahdollisia vääriä positiivisia UDP-tulvien havaitsemisia voitaisiin tulevaisuudessa ehkäistä valkolistaamalla tiettyjä sisäverkon IP-osoitteita tai portteja, jotta esimerkiksi sisäverkossa tapahtuvaa suoratoistamista ei tulkita palvelunestohyökkäykseksi, tai nostamalla ohjelmassa käytettävää kynnysarvoa. ICMP- ja TCP SYN-tulvien tunnistamisessa ohjelman ei pitäisi ilmoittaa vääristä positiivisista tuloksista tunnistusheuristiikkojen vuoksi; TCP SYN-FIN suhde ei ole normaalissa verkkoliikenteessä lähellä nollaa TCP-yhteyden symmetrisen luonteen takia, ja ICMP-pakettimäärät eivät yleensä nouse normaalissa verkkoliikenteessä korkeaksi lyhyellä aikavälillä.

Ohjelman kehityksen aikana huomattiin, että kaistanleveyttä ei voida käyttää tunnistusheuristiikan kynnysarvona, koska Scapy-kirjaston sniff-funktio ei saa havaittua kaikkia lähetettyjä paketteja. Kaistanleveys olisi ollut hyvä kynnysarvo volumetristen DoS-hyökkäystyyppien havaitsemiseen, sillä sen avulla oltaisiin voitu tarkemmin arvioida hyökkäyksen koko. Käytetyssä heuristiikassa on myös selkeitä heikkouksia. Esimerkiksi jos palvelunestohyökkäyksessä käytettyjen pakettien lähettäjän IP-osoite satunnaistetaan joka paketissa, hyökkäys jää tunnistamatta, sillä ohjelma tarkastelee liikennettä vain yhden IP-osoite-parin välillä.

Tulosten perusteella kynnysarvoja voitaisiin nostaa ainakin 3500 pakettiin 30 sekunnin aikana, millä testeissä käytetyt palvelunestohyökkäykset olisi myös havaittu ja todennäköisyys havaita normaali verkkoliikenne palvelunestohyökkäykseksi olisi pienempi.

Väärien positiivisten tulosten ehkäisemiseksi jatkossa voitaisiin hyödyntää normaalin verkkoliikenteen mallintamista ja kynnysarvojen automaattista määrittämistä. Verkkoliikenteen mallintaminen mahdollistaisi poikkeavuuksien

tunnistamisen vertaamalla verkkoliikennettä aikaisempaan esimerkiksi laitteen käyttöönoton aikaan havaittuun verkkoliikenteeseen. Mallintamisella voitaisiin parantaa luotettavuutta varsinkin tilanteissa, joissa laite lähettää normaalia verkkoliikennettä suurella kaistanleveydellä. Mallintamisen avulla ohjelma voi myös automaattisesti asettaa kynnysarvot hyökkäysten tunnistamiselle.

6.2. DDoS-työkalut

Testauksessa käytettiin itse kirjoitettuja DoS-työkaluja, jotka lähettävät samankaltaisia paketteja kuin bottiverkon saastuttama IoT-laite lähettäisi palvelunestohyökkäyksen aikana. Ohjelmaa olisi ollut hyvä testata myös oikealla saastuneen laitteen lähettämällä verkkoliikenteellä, mutta valitettavasti emme löytäneet tietokokonaisuuksia, jotka olisivat soveltuneet siihen.

Testissä 1 käytetty DNS-tulva-työkalun suorituskyky oli muihin työkaluihin verrattuna paljon heikompi, mutta riittävä, sillä työkalun lähettämät pakettimäärät olivat lähellä määrää, jonka ohjelma kykenee maksimissaan havaitsemaan. UDP-, TCP SYN- ja ICMP-tulva-työkalujen suorituskyky testaukseen oli myöskin riittävä, koska ohjelma ei saanut testeissä 2-4 havaittua kuin murto-osan lähetetyistä paketeista.

6.3. Scapy-kirjaston heikko suorituskyky

Johtuen Scapy-kirjaston sniff-funktion heikosta suorituskyvystä ohjelma saa kiinni huonoimmassa tapauksessa noin 1,6 % hyökkäyksessä lähetetyistä paketeista, mikä rajoittaa huomattavasti mahdollisten hyökkäysten tunnistamiseen tarkoitettujen menetelmien toteuttamista. Scapy listaakin verkkosivuillaan[1] tunnetuksi ongelmaksi osan paketeista jäävän havaitsematta, mikäli kuormitus on suuri. Jos ohjelma kykenisi Scapyn avulla tarkkailemaan kaikkia verkossa liikkuvia paketteja, kynnysarvot voisivat olla huomattavasti suurempia, mikä vähentäisi todennäköisyyttä tunnistaa normaali verkkoliikenne palvelunestohyökkäykseksi.

6.4. Jatkokehitys

Jatkokehityksen kannalta tärkeintä olisi parantaa pakettien tarkkailemisen suorituskykyä, jolloin ohjelma saisi ideaalitapauksessa kaikki paketit havaittua. Vaihtoehtoisesti Scapy-kirjaston sniff-funktion sijaan pakettien tarkkailuun voitaisiin käyttää esimerkiksi Wireshark- tai tcpdump-ohjelmia. Työssä käytetty Raspberry Pi 2 -tietokone oli hyvä testialusta ohjelman suorittamiseen, mutta ei suorituskyvyltään optimaalinen verkkoliikenteen valvomiseen käytettävänä tukiasemana. Ohjelman suorittaminen tehokkaammalla laitteella olisi yksi vaihtoehto paremman suorituskyvyn saamiseksi. Laitteena voisi toimia esimerkiksi ohjelman suorittamiseen räätälöity tehokkaampi langaton reititin tai analysoinnin voisi toteuttaa mahdollisesti pilvipohjaisessa ympäristössä.

[1] https://scapy.net/

Ohjelmaan voisi tulevaisuudessa lisätä muitakin menetelmiä tunnistaa eri DoS-hyökkäystyyppejä, kuten esimerkiksi HTTP-pohjaiset hyökkäykset, jotka ovat Kasperskyn [21] mukaan melko yleisiä, kattaen noin 8.8 % vuoden 2018 ensimmäisen vuosineljänneksen kaikista palvelunestohyökkäyksistä. Uusia DoS-hyökkäystyyppejä syntyy varsin tiheällä aikavälillä, joten ainakin yleisimmät hyökkäystyypit pitäisi pystyä tunnistamaan, mikä edellyttää tunnistusheuristiikan päivittämistä tulevaisuudessa.

Käyttäjäystävällisyyttä voitaisiin parantaa muun muassa laitteen tunnistamisella ja ilmoittamalla käyttäjälle hyökkäyksestä push-viesteillä. Laitteen tunnistaminen MAC- ja IP-osoitteen perusteella voi olla normaalille käyttäjälle haastavaa, joten laitteen tunnistaminen automaattisesti helpottaisi huomattavasti saastuneen laitteen tunnistamista. Tieto palvelunestohyökkäyksestä joudutaan erikseen lukemaan verkkosivulta, eikä laite lähetä minkäänlaista ilmoitusta siitä käyttäjälle. Laitteen käytettävyyden kannalta olisi parempi, mikäli laite lähettäisi jonkin varoituksen tai ilmoituksen käyttäjälle palvelunestohyökkäyksestä esimerkiksi käyttäjän puhelimeen.

7. AJANKÄYTTÖ

Suurin osa työskentelystä tehtiin tapaamalla yhdessä, joko jonkin ryhmän jäsenen luona tai internetin välityksellä ryhmäpuhelussa. Tapaamiset kestivät noin kahdesta neljään tuntia, joiden aikana työskenneltiin työn parissa. Taulukossa 8 on esitetty työhön käytetyt tunnit eri ryhmän jäsenten välillä.

Taulukko 8. Kandidaatintyöhön käytettyjen tuntien määrä ryhmän jäsenten välillä.

Nimi	Työhön käytettyjen tuntien määrä
Santeri Moberg	207
Joonas Hilke	203
Juuso Säärelä	197

8. YHTEENVETO

Palvelunestohyökkäyksien tunnistaminen ja torjuminen on hyvin ajankohtainen ja yleistyvä ongelma, hyökkäysten määrän kasvaessa huomattavasti joka vuosi. IoT-laitteet ovat houkuttava kohde pahantahtoisille palvelunestohyökkäyksiä lähettäville bottiverkoille, johtuen IoT-laitteiden heikosta tietoturvasta, minkä vuoksi tarvitaan erilaisia tietoturvaratkaisuja turvaamaan kodin IoT-laitteet. Bottiverkkojen kehittyessä uusia hyökkäystyyppejä ilmestyy jatkuvasti, mikä vaatii jatkuvaa kehitystyötä hyökkäysten torjumiseksi.

Työssä keskityttiin havaitsemaan langattomaan tukiasemaan yhdistynyt palvelunestohyökkäykseen osallistuva IoT-laite. Hyökkäysten havaitsemiseksi kehitettiin ohjelma, jota suoritetaan langattomassa tukiasemassa, johon IoT-laitteet ovat yhdistyneet. Ohjelman tarkoitus on havaita TCP SYN-, UDP-, DNS- ja ICMP-tulva-hyökkäykset tarkkailemalla ja analysoimalla tukiasemaan yhdistyneiden IoT-laitteiden verkkoliikennettä.

Ohjelman toimintaa testattiin simuloimalla edellämainittuja DoS-hyökkäystyyppejä itsetehdyillä DoS-työkaluilla ja suoratoistamalla videokuvaa VLC media player- ja Plex media server -ohjelmilla. Testeissä ohjelma kykeni havaitsemaan kaikki testauksessa käytetyt DoS-hyökkäystyypit, mutta ohjelman suorituskyky ei riittänyt havaitsemaan kaikkia hyökkäyksissä lähetettyjä paketteja. Lisäksi huomattiin, että ohjelma havaitsi VLC media playerin lähettämän UDP-verkkoliikenteen virheellisesti palvelunestohyökkäykseksi. Muilla hyökkäystyypeillä ei testeissä havaittu samanlaista ongelmaa.

9. LÄHTEET

[1] Atzori L, Iera A,& Morabito G (2010) The Internet of Things: A survey. Computer Networks 54(15): 2787-2805.
DOI: https://dx.doi.org/10.1016/j.comnet.2010.05.010

[2] Lund D, MacGillivray C, Turner V & Morales M (2014) Worldwide and regional internet of things (iot) 2014–2020 forecast: A virtuous circle of proven value and demand. International Data Corporation (IDC), Tech. Rep, 1.

[3] Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016 (2017). Gartner, Inc.
URL: https://www.gartner.com/newsroom/id/3598917 Vierailtu 27.3.2018

[4] Taneja M (2013) An analytics framework to detect compromised IoT devices using mobility behavior. Proc. 2013 IEEE International Conference on ICT Convergence (ICTC). Jeju, South Korea 38–43.
DOI: https://dx.doi.org/10.1109/ICTC.2013.6675302

[5] Oulun yliopiston langattoman tietoliikenteen 6G-tutkimus on valittu Suomen Akatemian rahoittamaksi tutkimuksen lippulaivaksi (2018). Oulun yliopisto.
URL: http://www.oulu.fi/yliopisto/node/52065 Vierailtu 10.10.2018

[6] Kopetz H (2011) Internet of Things. Real-Time Systems. Real-Time Systems Series. Springer, Boston, MA
DOI: https://dx.doi.org//10.1007/978-1-4419-8237-7_13

[7] Krebs B (2017) Dahua, Hikvision IoT Devices Under Siege. Krebs on Security URL:https://krebsonsecurity.com/2017/03/dahua-hikvision-iot-devices-under-siege/ Vierailtu 28.2.2018

[8] Fridge sends spam emails as attack hits smart gadgets (2014). BBC.
URL: http://www.bbc.com/news/technology-25780908 Vierailtu 1.2.2018

[9] Abomhara M & Køien G M (2015) Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. Journal of Cyber Security and Mobility 4(1): 65-88. DOI:
https://dx.doi.org/10.13052/jcsm2245-1439.414

[10] Babar S, Mahalle P, Stango A, Prasad N & Prasad R (2010) Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT). Recent Trends in Network Security and Applications. CNSA 2010. Communications in Computer and Information Science, vol 89: 420-429 Springer, Berlin, Heidelberg
DOI: https://dx.doi.org/10.1007/978-3-642-14478-3_42

[11] Q4 2017 State of the Internet / Security Report (2017). Akamai. URL: https://content.akamai.com/us-en-PG10413-q4-17-soti-security-report.html Vierailtu 25.2.2018

[12] Akita | Instant Privacy for Smart Homes. URL: https://www.kickstarter.com/projects/akita/akita-instant-privacy-for-smart-homes Vierailtu 4.2.2018

[13] F-Secure Sense. URL: https://www.f-secure.com/fi_FI/web/home_fi/f-secure-sense Vierailtu 4.2.2018

[14] Cujo Ai. URL: https://www.getcujo.com/ Vierailtu 4.2.2018

[15] Lau F., Rubin S. H., Smith M. H., Trajkovic L. (2000) Distributed denial of service attacks. 2000 IEEE International Conference on Systems, Man and Cybernetics. Nashville, TN, 2275-2280
DOI: https://dx.doi.org/10.1109/ICSMC.2000.886455

[16] Thomas D R, Clayton R & Beresford A R (2017) 1000 days of UDP amplification DDoS attacks. 2017 APWG Symposium on Electronic Crime Research (eCrime). Scottsdale, AZ, USA, 79-84.
DOI: https://dx.doi.org/10.1109/ECRIME.2017.7945057

[17] Rossow C (2014) Amplification Hell: Revisiting Network Protocols for DDoS Abuse. NDSS Symposium 2014. San Diego, CA, USA
DOI: https://dx.doi.org/10.14722/ndss.2014.23233

[18] Palvelunestohyökkäysten tekniikkaa puolustajille (2016). Viestintävirasto, kyberturvallisuuskekus.
URL:
https://www.viestintavirasto.fi/attachments/tietoturva/Ohje_3_2016_liite_1_Palvelunestohyokkaysten_tekniikkaa_puolustajille.pdf Vierailtu 10.10.2018

[19] Bekerman D (2017) Gbps vs. pps vs. rps DDoS: On Volumetric, Protocol and Application Layer Attacks. Incapsula. URL:
https://www.incapsula.com/blog/gbps-pps-rps-ddos-attacks.html Vierailtu 10.10.2018

[20] Application Layer DDoS attacks.
URL: https://www.cloudflare.com/learning/ddos/application-layer-ddos-attack/ Vierailtu 10.10.2018

[21] Khalimonenko A, Kupreev O, Ilganaev K (2018) DDoS attacks in Q4 2017. Kaspersky Lab. URL: https://securelist.com/ddos-attacks-in-q4-2017/83729/ Vierailtu 24.9.2018

[22] Khalimonenko A, Kupreev O, Badovskaya E (2018) DDoS attacks in Q1 2018. Kaspersky Lab. URL: https://securelist.com/ddos-report-in-q1-2018/85373/ Vierailtu 11.6.2018

[23] Ibragimov T, Kupreev O, Badovskaya E, Gutnikov A (2018) DDoS attacks in Q2 2018. Kaspersky Lab. URL: https://securelist.com/ddos-report-in-q2-2018/86537/ Vierailtu 13.9.2018

[24] Korhonen S (2016) Verkkoisku kylmensi useita taloja suomessa - ES: "Lämmitys ja kuuma vesi pois päältä". Tivi.
URL: https://www.tivi.fi/Kaikki_uutiset/verkkoisku-kylmensi-useita-taloja-suomessa-es-lammitys-ja-kuuma-vesi-pois-paalta-6597180 Vierailtu 25.2.2018

[25] Kumar M (2016) DDoS Attack Takes Down Central Heating System Amidst Winter In Finland. The Hacker News.
URL: https://thehackernews.com/2016/11/heating-system-hacked.html Vierailtu 25.2.2018

[26] Uzunovic A (2016) Wikileaks Releases DNCLeak2; Suffers Massive DDoS Attack. HackRead.
URL: https://www.hackread.com/wikileaks-dncleak2-suffers-massive-ddos-attack/ Vierailtu 25.2.2018

[27] Mansfield-Devine S (2015) The growth and evolution of DDoS. Network Security 2015 Issue 10, 13-20.
DOI: https://dx.doi.org/10.1016/S1353-4858(15)30092-1

[28] Akamai Releases Findings Of Increased Attacks And More Aggressive Tactics From DD4BC Extortionist Group (2015). Akamai.
URL: https://www.akamai.com/us/en/about/news/press/2015-press/akamai-plxsert-releases-findings-on-dd4bc-bitcoin-attack-tactics.jsp Vierailtu 10.10.2018

[29] Kottler S (2018) February 28th DDoS Incident Report. GitHub Engineering.
URL: https://githubengineering.com/ddos-incident-report/ Vierailtu 29.3.2018

[30] Kumar M (2018) 1.7 Tbps DDoS Attack — Memcached UDP Reflections Set New Record. The Hacker News.
URL: https://thehackernews.com/2018/03/ddos-attack-memcached.html Vierailtu 27.8.2018

[31] Townsend K (2018) Largest Ever 1.3Tbps DDoS Attack Includes Embedded Ransom Demands. Securityweek.
URL: https://www.securityweek.com/largest-ever-13tbps-ddos-attack-includes-embedded-ransom-demands Vierailtu 27.8.2018

[32] Majkowski M (2018) Memcrashed - Major amplification attacks from UDP port 11211. Cloudflare. URL: https://blog.cloudflare.com/memcrashed-major-amplification-attacks-from-port-11211/ Vierailtu 29.3.2018

[33] Nychis G, Sekar V, Andersen D G, Kim H & Zhang H (2008) An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. 8th ACM

SIGCOMM conference on Internet measurement. Vouliagmeni, Greece, 151-156. DOI: https://dx.doi.org/10.1145/1452520.1452539

[34] Makrushin D (2017) The cost of launching a DDoS attack. Securelist. URL: https://securelist.com/the-cost-of-launching-a-ddos-attack/77784/ Vierailtu 11.3.2018

[35] Oikarinen J & Reed D (1993) Internet relay Chat Protocol. RFC 1459. IETF. URL: https://tools.ietf.org/html/rfc1459

[36] McCarty B (2003) Botnets: big and bigger. IEEE Security & Privacy 99(4): 87-90. DOI: https://dx.doi.org/10.1109/MSECP.2003.1219079

[37] Gu G, Zhang J, & Lee W (2008) BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. Proceedings of the 15th Annual Network and Distributed System Security Symposium.

[38] Grizzard J B, Sharma V, Nunnery C, Kang B B, & Dagon D (2007) Peer-to-Peer Botnets: Overview and Case Study. Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets 1-1.

[39] Attack of Things! (2016). Level 3 Threat Research Labs. URL: https://www.netformation.com/our-pov/attack-of-things-2/ Vierailtu 21.2.2018

[40] Wang A, Liang R, Liu X, Zhang Y, Chen K & Li J (2017) An Inside Look at IoT Malware. International Conference on Industrial IoT Technologies and Applications, Wuhu, China, 176-186. DOI: https://dx.doi.org/10.1007/978-3-319-60753-5_19

[41] Oltermann P (2017) Briton admits to cyber-attack on Deutsche Telekom. The Guardian. URL: https://www.theguardian.com/world/2017/jul/21/briton-admits-to-cyber-attack-on-deutsche-telekom-court Vierailtu 25.2.2018

[42] BrickerBot PDoS Attack: Back With A Vengeance (2017). Radware. URL: https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-back-with-vengeance/ Vierailtu 10.3.2018

[43] Dr Cyborkian a.k.a. janit0r (2017) Internet Chemotherapy. URL:https://github.com/JeremyNGalloway/mod_plaintext.py/blob/master/Internet%20Chemotherapy Vierailtu 27.3.2018

[44] Reaper-bottiverkko saastuttaa IoT-laitteita (2017). Viestintävirasto. URL: https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2017/11/ttn20171711011452.html Vierailtu 25.2.2018

[45] Ye G (2017) IoT_reaper: A Rappid Spreading New IoT Botnet. Netlab 360. URL: http://blog.netlab.360.com/iot_reaper-a-rappid-spreading-new-iot-botnet-en/ Vierailtu 25.2.2018

[46] Day J D & Zimmermann H (1983) The OSI reference model. Proceedings of the IEEE 71(12): 1334-1340.
DOI: http://dx.doi.org/10.1109/PROC.1983.12775

[47] https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml Vierailtu 11.6.2018

[48] Transmission control protocol (1981). RFC 793. IETF.
URL: https://tools.ietf.org/html/rfc793 Vierailtu 11.6.2018

[49] Postel J (1980) User Datagram Protocol. RFC 768. IETF.
URL: https://tools.ietf.org/html/rfc768 Vierailtu 11.6.2018

[50] Postel J (1981) Internet control message protocol. RFC 792. IETF.
URL: https://tools.ietf.org/html/rfc792 Vierailtu 11.6.2018

[51] Harris B & Hunt R (1999) TCP/IP security threats and attack methods. Computer Communications 22(10): 885-897.
DOI: https://doi.org/10.1016/S0140-3664(99)00064-X

[52] Ansari S, Rajeev S G & Chandrashekar H S (2003) Packet sniffing: a brief introduction. IEEE Potentials 21(5): 17-19.
DOI: http://dx.doi.org/10.1109/MP.2002.1166620

[53] Belenguer J & Calafate C T (2007) A low-cost embedded IDS to monitor and prevent Man-in-the-Middle attacks on wired LAN environments. The International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007). Valencia, Spain, 122-127.
DOI: https://doi.org/10.1109/SECUREWARE.2007.4385321

[54] Wang H, Zhang D & Shin K G (2002) Detecting SYN flooding attacks. Proc. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. New York, NY, USA, 1530-1539.
DOI: https://doi.org/10.1109/INFCOM.2002.1019404